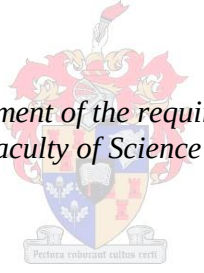
Compounding a class of Rayleigh distributions: an objective Bayesian approach

by
Renier van Rooyen



Thesis presented in fulfilment of the requirements for the degree of Master of Science in the Faculty of Science at Stellenbosch University

Supervisor: Prof Paul J Mostert

December 2015

Declaration

By submitting this thesis/dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

December 2015

"To those who do not know mathematics it is difficult to get across a real feeling as to the beauty, the deepest beauty, of nature. If you want to learn about nature, to appreciate nature, it is necessary to understand the language that she speaks in."

Richard Feynman

UNIVERSITY OF STELLENBOSCH

Abstract

Compounding a class of Rayleigh distributions: an objective Bayesian approach

by Renier van Rooyen

In this work, Bayesian estimation in the context of parametric survival analysis is considered. A class of models derived by compounding and generalising the Rayleigh distribution is regarded. These models are well suited to survival analysis settings where the hazard rate is characterised by a sharp increase over time. An objective Bayesian approach is followed, whereby non-informative prior distribution selection leads to the use of the Jeffreys, the reference and the probability matching priors. Bayesian point estimators are derived using two symmetric loss functions, namely absolute error and squared error, as well as two asymmetric loss functions, namely linear exponential and general entropy. The resulting models and estimators are showcased in a simulation study by generating right censored lifetime data from the various compound models and utilising the Metropolis-Hastings algorithm to draw realisations from the corresponding posterior distributions, since closed-form expressions for these cannot be found. Obtaining the Fisher information plays a crucial part in deriving the non-informative priors. In cases where it cannot be analytically evaluated, an adaptive quadrature routine is used for the numerical approximation of some of the elements in the Fisher information. An application to data sets from practice concludes the exposition of the compound Rayleigh models of interest.

UNIVERSITEIT VAN STELLENBOSCH

Opsomming

Compounding a class of Rayleigh distributions: an objective Bayesian approach

deur Renier VAN ROOYEN

In hierdie tesis, word Bayes-beraming beskou in die konteks van parametriese oorlewingsanalise. 'n Klas modelle wat afgelei is deur samestelling en veralgemening van die Rayleigh-verdeling, word beskou. Hierdie modelle is toepaslik in oorlewingsanalisescenarios waar die gevaarfunksie beskryf word deur 'n skerp toename oor tyd. 'n Objektiewe Bayes-benadering word gevolg en die toepaslike keuse van nie-inligtinggewende prior-verdelings lei na die gebruik van die Jeffreys-, die verwysings- en die waarskynlikheidspassende priors. Bayes puntberamers word afgelei met inagneming van twee simmetriese verliesfunksies, naamlik absolute fout en kwadratiese fout, sowel as twee asimmetriese verliesfunksies, naamlik lineêr eksponensieel en algemene entropie. Die gevolglike modelle en beramers word ten toon gestel in 'n simulasiestudie deur regsgesensoreerde leeftyd-data te genereer vanuit die verskeie saamgestelde modelle en dan die Metropolis-Hastings algoritme te gebruik om realiserings vanuit die ooreenstemmende posterior-verdelings te verkry, aangesien oplossings vir hierdie funksies nie in geslote vorm gevind kan word nie. Die bepaling van die Fisher-inligting speel 'n kardinale rol in die afleiding van die nie-inligtinggewende priors. In gevalle waar dit nie analities evalueer kan word nie, word 'n aanpassende kwadratuurroetine gebruik vir die numeriese benaderings van sommige elemente in die Fisher-inligting. Laastens word die uiteensetting van die saamgestelde Rayleigh modelle afgesluit deur die toepassing op twee datastelle uit die praktyk.

Acknowledgements

I would like to express my sincerest gratitude to the DST-NRF Centre of Excellence in Epidemiological Modelling and Analysis (SACEMA) and the Institute of Applied Statistics (IAS) for providing guidance and financial support, and accordingly making my M.Sc. studies possible. (Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to SACEMA or the IAS.)

I want to thank Professor Paul Mostert unreservedly for the expertise, continuous help, motivation, encouragement and enormous amounts of patience. I also want to thank Dr. Martin Nieuwoudt as well as others at SACEMA for many ideas and enthusiasm. Lastly my friends and loved ones, especially my parents, Club 84 and Ilani, for inspiring me to never give up.

Contents

List of Figures

List of Tables

Abbreviations

AE	absolute error
CDF	cumulative distribution function
CRE	compound Rayleigh with respect to exponential distribution
CRG	compound Rayleigh with respect to Gamma distribution
DIC	deviance information criterion
GCRE	generalised compound Rayleigh with respect to exponential distribution
GCRG	generalised compound Rayleigh with respect to Gamma distribution
GE	general entropy (also abbreviated as GENT in some cases)
iid	independent and identically distributed
LINEX	linear exponential
log	logarithm
ln	natural logarithm
MCMC	Markov chain Monte Carlo
MAE	mean absolute error
MSE	mean squared error
PDF	probability density function
SE	squared error

Symbols

$P(\cdot)$	probability
$P(\cdot\|\cdot)$	conditional probability
$\mathrm{E}_\theta[\cdot]$	expected value with respect to distribution of θ
$\mathrm{V}_\theta[\cdot]$	variance with respect to distribution of θ
$f(\cdot)$	probability density function (PDF)
$F(\cdot)$	cumulative distribution function (CDF)
$\mathcal{L}(\cdot)$	likelihood function
$S(\cdot)$	survival function
$h(\cdot)$	hazard rate
T	random survival time variable
t	arbitrary survival time measurement
$\mathbf{t}$	set of survival time data
n	sample size
d	number of non-censored survival times in sample
δ	proportion of non-censored survival times in sample
$\mathcal{I}_F$	Fisher information (matrix)
$[I]_{i,j}$	element row i and column j of matrix I
$\|I\|$	determinant of matrix I
e, $\exp(1)$	Euler's number (2.71828...)
π	ratio of a circle's circumference to its diameter (3.14159...)
$U(0,1)$	a random variable uniformly distributed between 0 and 1

$\pi(\cdot)$	arbitrary prior distribution
$\pi(\cdot\|\cdot)$	arbitrary posterior distribution
$\pi_{\text{jeff}}(\cdot)$	Jeffreys' prior
$\pi_{\text{ref}}(\cdot)$	reference prior
$\pi_{\text{PM}}(\cdot)$	probability matching prior

$L_{\text{AE}}(\cdot)$	absolute error loss
$L_{\text{SE}}(\cdot)$	standard error loss
$L_{\text{LNX}}(\cdot)$	linear exponential loss
$L_{\text{GE}}(\cdot)$	general entropy loss

Chapter 1

Introduction

In a very simplified sense, Statistics can be seen as a tool for deriving usable information from a given set of data. The scientific method compels us to do this derivation in the most objective of ways, thus the field finds its roots in the universal language of logic – mathematics. The Greek historian Herodotus is reported to have first discussed basic decision theory with the Persians in 500 B.C., but it is only in the 16th century that the mathematical foundations were laid (Jaynes, 1984). Around this time, probability theory, the framework on which statistical inference is now built on, quickly excelled and lead to James Bernoulli publishing *Ars Conjectandi* (*The Art of Conjecture*) in 1713. In this work, he set out to give a mathematical representation of hypotheses, focusing on cases that are equally probable. He was also the first to show and prove the mathematical connection between frequency and probability. If M denotes the possible ways in which a proposition A can be true and N denotes the total number of outcomes in the hypothesis space, consider an experiment in which it was found that A was true m times out of n independent observations. Bernoulli showed that as n becomes large, $\frac{m}{n}$ will be close to $\frac{M}{N}$, the true probability of A. Today, this is known as the weak law of large numbers and also, in a way, laid the basis for what is now called "frequentist" statistical inference. The frequentist way of thinking revolves around the expected outcome of an experiment that is repeated a very large number of times and many analytical techniques and methodologies have been devised around this central idea, to aid in understanding data and drawing subsequent conclusions.

Not long after Bernoulli, a competing paradigm of statistical ideas arose from the work of Reverend Thomas Bayes and the great mathematician Pierre Simon Laplace. The theorem they devised laid the foundation for a whole new approach to data analysis, which has recently been gaining popularity and is being employed widely in modern applied Statistics (Poirier, 2006). Bayesian methods now present statisticians with novel ways to approach old data modelling problems for which solutions was conceptually frequentist in nature.

One of these interesting classes of data emerges in studies where the times to some event of interest are measured. This type of data is in abundance in medical studies, but also abound in fields of engineering where reliability is studied. The branch of Statistics that deals with these data sets is commonly termed *survival* or *lifetime* analysis and the measurements called *survival times* or *lifetime data*.

This thesis investigates Bayesian modelling in the context of survival analysis. A specific class of parametric models, emanating from the Rayleigh distribution, is considered, and estimators are derived and compared using an array of loss functions and objective, or "non-informative", prior distributions. The primary goal is to assess the performance of these different estimators and priors, as well as compare them when appropriate, within each model. This work is a continuation of research conducted by Mostert et al. (1999) and Bekker et al. (2000). They studied the Rayleigh distribution for survival analysis of cancer data in the Bayesian framework, using compounding and generalisation to derive modified models. Their research is extended here by considering a wider array of compounding distributions and loss functions, but more importantly, by allowing all unknown model parameters to remain continuous in its specification instead of discretisation to a finite set of values. Earlier work on similar models was performed by Greenwich (1992) and Abdel-Ghaly and Attia (1993), but they do not consider Bayesian methodology. More recently, Guure et al. (2012) studied a similar type of model with a Bayesian approach, but with only one non-informative prior and otherwise limited in scope.

The important concepts of survival analysis and Bayesian statistics relevant to this thesis are discussed in Chapter 2. Chapter 3 outlines a literature review of the Rayleigh distribution and its modifications in the survival analysis context, as well as defining the modifications to this model which will be considered here, namely compounding and

generalisation. Chapter 4 presents the simulation study and results of the compound Rayleigh models and Chapter 5 follows in similar vein with the simulation study for the Rayleigh models which are both compounded and generalised. The models, priors and estimators derived are applied to two data sets in Chapter 6 and Chapter 7 concludes with a final discussion of the overall findings.

Chapter 2

Survival analysis and the Bayesian method

In this chapter, a brief outline of survival analysis is provided, focusing on the main concepts, important parameters and especially the parametric approach to performing survival analysis. Thereafter, parameter estimation with the Bayesian method is discussed in Section 2.2. An overview of the whole method, from prior specification to posterior distribution simulation, is given, with a focus on relevant topics for the current work, namely non-informative priors, loss functions and approximation of the Fisher information. The chapter concludes on a philosophical note, motivating some of the modelling choices undertook here.

2.1 Concepts in survival analysis

The field of survival analysis describes the inferential methodologies regarding a specific type of data that deals with the time to an event of interest. A unique feature of lifetime data is that not all times are completely and exactly observed. For example, some individuals may not have experienced the event by the end of the study and needs to be cut off at a certain time. Alternatively, an individual might withdraw from the study for unrelated reasons. These observations are then called censored and contains only partial information relating to the event of interest. The type of censoring regulates how this partial data are incorporated into the likelihood. In the case of these examples,

or any scenario in which times are observed before a predetermined time, the term *right censoring* is used. One can also distinguish between different right censoring schemes.

- *Type I censoring*: the end time of the study is predetermined and all remaining individuals are censored after that time.
- *Type II censoring*: the amount of events observed is predetermined and once that amount has been reached, all remaining individuals in the study are censored.
- *Random censoring*: individuals' event times and censored times are statistically independent, and censoring occurs when the former happens after the latter.

Other types of censoring exist, such as left censoring, where individuals may experience the event before the study, or interval censoring, where the time of the event is only known to lie in some interval, but these fall beyond the scope of this thesis. A detailed overview can be found in Klein and Moeschberger (2003).

Survival analysis techniques have been employed in many fields besides medical research, such as epidemiology, engineering and economics. Trivial examples include investigating the time to recovery of some illness or operation, the time to outbreak of some disease or the time to failure of a mechanical component. The data obtained from these studies all share a commonality in that it is consisted of each individual's time measurement to some event, subject to censoring. Survival analysis is mainly concerned with parameters that regard the distribution of the lifetimes, such as the *mean lifetime* and the *mean residual lifetime* which respectively quantifies the expected survival time and expected future survival time given survival up to a specified age. Two very important parameters relevant to the current work, the survival function and the hazard rate, are discussed in Section 2.1.1, followed by a brief overview of the ways in which they can be estimated in Section 2.1.2.

2.1.1 Important parameters in a survival analysis

In order to discuss more technical aspects of survival analysis, it is important to review some cornerstones of distribution theory. In general, one can think of an observation t as a realisation of a random variable T, which is distributed according to a probability mechanism f. The function f usually depends on known or unknown parameters and

depending on the value of these parameters, $f(t)$ equates to the probability of observing t. Accordingly, the summation or integral of f over all possible values of T equals one. When this holds, f is called a probability density function (PDF) and its integral (in the continuous case) the cumulative distribution function (CDF) F, such that

$$F(t) = \int_{-\infty}^{t} f(x)dx = P(T \leq t).$$

The nature of a survival time measurement t is such that it is both continuous and non-negative, in other words $t > 0$.

Considering $F(t)$, the cumulative distribution function for a random survival time, the survival function can consequently be defined as the probability of an individual experiencing the event of interest beyond time t.

$$\begin{aligned} S(t) &= P(T > t) \\ &= 1 - F(t). \end{aligned} \tag{2.1}$$

This is akin to the probability of an individual surviving up to (and including) time t.

A second key parameter is the hazard rate (sometimes called the "failure rate" or the "intensity function"). This is the rate at which events happen at time t and is defined as

$$\begin{aligned} h(t) &= \frac{f(t)}{S(t)} \\ &= -\frac{\partial}{\partial t} \ln S(t) \\ &= \lim_{dt \to 0} \frac{P(t \leq T < t + dt | T > t)}{dt}. \end{aligned} \tag{2.2}$$

From the definition in (2.2) it can be seen that the hazard rate at time t approximates the probability of an individual experiencing the event in the next instant, conditional upon survival up to time t. The hazard rate is of paramount importance, since knowledge regarding the way the probability of experiencing the event changes over time is valuable to most applications of survival analysis.

The likelihood function is not a parameter in survival analysis, but it is crucial to many methods of parameter estimation, such as the Bayesian methods employed here and described in Section 2.2, and it is thus important to take note of. The likelihood quantifies

the probability of a given outcome as a function of the model parameters. Consider the probability model of the survival times f as above and assume it is dependent on some parameter(s) θ. Furthermore, consider a sample of n independent and identically distributed (iid) survival times, denoted by $\mathbf{t} = (t_1, t_2, \ldots, t_n)$. Generally, the likelihood then follows as the product of the probabilities of the observations, such that

$$\mathcal{L}(\theta|\mathbf{t}) = \prod_{i=1}^{n} f(t_i|\theta).$$

In a survival analysis, however, some of the data may be censored and this partial information leads to a structural change in the likelihood function. Dellaportas and Wright (1991) present a way to obtain the likelihood in the presence of a right censored sample. Suppose that $\mathbf{t}$ is ordered such that $t_1 < t_2 < \ldots < t_d$ are events and $t_{d+1} < t_{d+2} < \ldots < t_n$ are right censored observations. Then, using (2.1), the likelihood function becomes

$$\mathcal{L}(\theta|\mathbf{t}) = \frac{n!}{n-d} \prod_{i-1}^{d} f(t_i|\theta) \prod_{j=d+1}^{n} S(t_j), \tag{2.3}$$

providing a way to utilise the incomplete information inherent in the censored observations in an analytical manner.

2.1.2 Different approaches to parameter estimation

Broadly speaking, approaches to estimate the important parameters in a survival analysis can be classified into three categories depending on the specification of the underlying model. Note that the term "parametric" here refers to the model parameter θ of a probability distribution, not the survival and hazard functions discussed in the previous section.

The non-parametric approach makes no distributional assumptions about the data and estimates the important parameters directly. This is a common and simple way to perform a survival analysis and is often used to get an idea of the form of the survival function. One immensely popular method uses the Kaplan-Meier estimator, also known as the product limit estimator, which can be regarded as a type of empirical distribution function for censored data (Kaplan and Meier, 1958). This leads to a stepwise survival function estimate. An alternative method, that of Nelson-Aalen, estimates the hazard rate non-parametrically, but usually attains very similar results as the Kaplan-Meier

estimator, and in fact, can be shown be to asymptotically equivalent (Colosimo et al., 2002). It can be argued that the inferential scope of non-parametric survival analysis methods are limited and furthermore, the discrete nature of the estimators are not well suited for predictive purposes, especially with little data (Berliner and Hill, 1988).

The second approach is semi-parametric in nature. This not only improves the predictive abilities of the survival model over the non-parametric approach, but also allows for inclusion of covariates as part of a regression, such that different attributes of groups, for which a comparison might be desired, can be controlled for (Klein and Moeschberger, 2003). A common choice for this type of modelling is the proportional hazards model of Cox, where the regression is based around an arbitrary baseline hazard rate (Cox, 1972). This baseline hazard $h_0(t)$ is non-parametric in nature, but used in the formulation of the regression model

$$h(t|\mathbf{z}) = h_0(t)e^{\mathbf{z}\boldsymbol{\beta}},$$

where $\mathbf{z}$ is a vector of covariates and $\boldsymbol{\beta}$ the corresponding vector of regression coefficients. The semi-parametric survival approach is advantageous, especially when the form of the hazard rate is not of prime importance, since it avoids the need to make assumptions about the underlying distribution of the data.

The final approach, parametric survival analysis, assumes that the survival times are distributed according to a fully parametrised probability distribution $f(t|\theta)$. Hereby, the forms of the survival and hazard functions can be explicitly derived in an analytical manner. The inference then revolves around estimating the parameters of the chosen distribution. The assumption of an underlying model is substantial, as it influences the entire analysis that follows, thus this decision should be made with care. However, there are cases where survival data shows patterns that tightly resemble a parametric probability distribution, or where characteristics of the rate at which the event of interest happens can be reconciled with the form of a specific hazard rate function. Some examples regarding the Rayleigh distribution is discussed in Section 3.1. Additionally, parametric survival analysis opens up a wide inferential scope, with the benefit of obtaining the exact form of the hazard rate, one of the most insightful outcomes. Alternatively, where it is sensible to do so, parametric models can also be derived by first selecting the hazard rate function (Martz and Waller, 1982).

For the remainder of this thesis, parametric survival models will be considered and in particular, modified Rayleigh distributions will be assumed to be the probability generating mechanisms. The parametric approach is motivated on philosophical grounds in Section 2.3. This approach is also in accordance with the decision to employ the Bayesian paradigm for estimation of the model parameters. Bayesian statistics and its relevance to the current work is the topic of the remainder of this chapter.

2.2 Overview of the Bayesian method

Thomas Bayes is described as being a mysterious clergyman who attracted attention only when his work on probability theory was published posthumously by one of his peers. The incredibly famous theory that carries his name can be stated in terms of two independent events, say A and B, such that (Bayes and Price, 1763)

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}.$$

Even though Bayes' name is attributed to the paradigm to which the law lead, it was Laplace who generalised the formula in 1773, apparently unaware of the earlier work by Bayes (Stigler, 1986). He modelled the uncertainty surrounding the parameter(s) θ of a parametric model through a probability distribution, called the *prior distribution* π, on the set of possible parameter values Θ, leading to

$$\pi(\theta|\mathbf{t}) = \frac{\mathcal{L}(\theta|\mathbf{t})\pi(\theta)}{\int_\Theta \mathcal{L}(\theta|\mathbf{t})\pi(\theta)d\theta}, \tag{2.4}$$

which is referred to as the *posterior distribution* of θ conditional on the data $\mathbf{t}$. The likelihood function $\mathcal{L}$ was defined in Section 2.1.1 and will be considered in more detail in Section 2.2.1. Bayesian methodology thus necessitates analyses *given the available data* and moreover, implicates a solid probabilistic framework in which to study any aspects of the model parameters, such as its moments, variance and quantiles.

Interestingly, Bayesian statistics was fairly popular until the early 20th century, when Fisher and Neyman started to develop the idea of confidence intervals and the resulting formalisation of statistics based on asymptotic theory (Jaynes, 1984). This gave rise to the frequentist approach, which have dominated modern day data analysis, because the

inherent assumptions and asymptotic reasoning lead to simplicity in the derivation and application of estimators. In contrast, a substantive problem with Bayes' law is that the integral in (2.4) often leads to mathematical difficulties, thus a closed-form solution to the posterior probability distribution cannot always be found. However, there has been a recent uprising in ways to obtain the posterior. One example is the use of clever simulation algorithms, such as Markov chain Monte Carlo (MCMC) methods. The heavy computational burden of these algorithms presents no problem for modern computers. The influx of these methods alleviate many of the mathematical difficulties inherent in Bayesian statistics and consequently, the use of the Bayesian method have become much more prominent in recent years, while applications have grown increasingly wider, as discussed by Beaumont and Rannala (2004), Poirier (2006) and Andrews and Baguley (2013), amongst others.

In Section 2.2.1, a description about the core concepts of a Bayesian analysis will be given in general terms. Section 2.2.2 will consider the formulation of Bayesian estimators, as well as their derivation under the loss functions used in this thesis. Sections 2.2.3 and 2.2.4 will respectively discuss the Fisher information and the non-informative priors of interest. Section 2.2.5 concludes with an overview of how MCMC is used to draw samples from posterior distributions.

2.2.1 Updating prior information to the posterior distribution

Consider a probability distribution f and assume it is the model from which a data set, denoted $\mathbf{t}$, is generated. This model is characterised by a parameter (or parameters) θ. The main objective of analyses is inference regarding θ. Traditionally, results are comprised of a point estimate of θ, with uncertainty summarised in the form of error measures or confidence intervals. Furthermore, decisions about θ or functions of θ can be investigated with hypothesis tests. Bayesian methodology tackles the uncertainty regarding the parameter in a completely different manner. By assuming that θ is stochastic in nature, with corresponding probability distribution π, the focus of inference now switches to obtaining the posterior distribution of the parameter given the data. It should be emphasised here that the assumption regarding the stochasticity of θ is predominantly utilitarian instead of factual. In many cases, one can rightfully debate that a random parameter is insensible. However, it is argued that assigning a probability

distribution is the most efficient and useful way of summarising information about the parameter as well as dealing with its uncertainty and additionally, this is necessary if we want a mathematically rigorous approach of conditioning on the data (Robert, 2007).

The posterior distribution is obtained using Bayes' theorem as given in equation (2.4) and accordingly, a broad range of conclusions related to θ can be drawn based on this distribution. More specifically, $\pi(\theta|\mathbf{t})$ emerges as a (scaled) product of the prior distribution and the likelihood function $\mathcal{L}(\theta|\mathbf{t})$, defined in Section 2.1.1. Even though it is not actually a conditional probability distribution, the way the likelihood is denoted emphasises the dependence on a specific data set. The likelihood is meant to represent how likely certain values of θ are in light of given observations of the random variable. It is also the cornerstone of an important, albeit controversial, rule in statistical estimation theory – the likelihood principle. This rule states that all the information in the sample required for inference about θ is contained within the likelihood function. There has been an ongoing debate about the validity of the likelihood principle (Hill, 1987), but even the term "information" can be an elusive concept, a topic investigated in Section 2.2.4. Generally, inference using Bayesian statistics adhere to the likelihood principle, since the posterior is dependent on the data only through the likelihood function $\mathcal{L}$.

Maximum likelihood estimation, a very popular technique for statistical inference, considers the maximisation of the likelihood function in order to obtain an estimate

$$\hat{\theta}_{\mathrm{ML}} = \arg\sup_{\theta} \mathcal{L}(\theta|\mathbf{t}).$$

This approach can be classified at the edge between frequentist and Bayesian paradigms and is intuitive, since it appears that the probability of occurrence of the given data is maximised. However, it lacks any formal probabilistic framework, since no distribution is assigned to θ, and as a result one has to rely on asymptotic theory for measures of uncertainty regarding the estimate. Other problems with this approach amounts to possible mathematical complexity, as well as unstable behaviour for small sample sizes (Robert, 2007).

With the posterior distribution as its focal point, the Bayesian statistician uses given data in a coherent way when performing inference. The information about the parameter is extracted from the data and used to update prior belief. With each new data point, the initial posterior distribution can be considered as a prior to be updated with

the additional observation. Once the posterior has been obtained, it can then be used to explore a variety of estimators, such as the mean or median, and additionally, judge their performance with the variance. Confidence regions also emerge intuitively in various forms. Bayesian *credible intervals* are constructed from quantiles of the posterior distribution (Eberly and Casella, 2003). For some significance level ψ, a $100(1-\psi)\%$ credible interval is a subset $\mathcal{C}$ of the domain Θ, such that

$$\int_{\mathcal{C}} \pi(\theta|\mathbf{t})d\theta = 1-\psi.$$

In other words, the $(\frac{\psi}{2})^{\text{th}}$ and $(1-\frac{\psi}{2})^{\text{th}}$ quantiles will respectively form the lower and upper bounds of a $100(1-\psi)\%$ credible interval. A second variety of Bayesian intervals are so-called highest posterior density (HPD) intervals, an approach more suited to asymmetric posterior distributions, but much more costly computationally (Chen and Shao, 1999). This is a special case of credible intervals that focuses on areas in the domain Θ corresponding to the highest posterior probabilities, as the name suggests. $100(1-\psi)\%$ HPD intervals are defined as the subset $\mathcal{C} = \{\theta : \pi(\theta|\mathbf{t}) \geq z^*\}$, where

$$z^* = \arg\max_z \int_{\theta:\pi(\theta|\mathbf{t})\geq z} \pi(\theta|\mathbf{t})d\theta = 1-\psi.$$

In the current work, only the formulation in terms of quantiles are used. Credible intervals allow one to state that a bounded region of the domain contains θ with a probability of $100(1-\psi)\%$. This also allows for probabilities to be assigned elegantly to hypotheses in statistical testing settings, in contrast with the frequentist approach where asymptotic arguments involving unobserved data need to be used.

At this point in the discussion of the Bayesian method, two questions remain: the first and final steps of the inferential process. Once the posterior is obtained, a method for choosing an estimator for the parameter based on this distribution is required. The derivation of so-called *Bayesian estimators* will be discussed in Section 2.2.2.

This leaves one fundamental aspect of a Bayesian analysis to be discussed. The prior distribution can have a large effect on the form of the posterior, thus it is worthwhile to consider its choice and derivation thoroughly. In layman's terms, one of the selling points of the Bayesian method is the ability to incorporate previous knowledge or information that may be available about a problem domain. Even though this is both true and

a very useful feature, translating knowledge into the rigorous language of Statistics is no easy task. Sometimes information is readily available from literature and then it is straightforward to use distributions and parameters estimated in previously conducted studies as the prior distribution for the current study. In most cases, however, prior domain knowledge may not be available, or may be in the form of speculation at best. Unfortunately, this subjectivity has been a major source of criticism and has plagued the Bayesian method by preventing widespread implementation (Gelman, 2008). It is one of the outcomes of this thesis to show that these causes of concern are unwarranted and that Bayesian statistics can effectively be applied objectively.

One way in which in which these problems have been addressed is by choosing a prior with attractive mathematical properties. A conjugate prior for a likelihood is when the prior and posterior are part of the same family of distributions. They are then also known as conjugate distributions. Conjugate priors are mainly used for their convenience and the consequent simplicity and transparency of the resulting analysis (Gelman et al., 2004). A detailed exploration into this topic is beyond the scope of the current work.

Rather than choosing priors based on convenient mathematical properties, the focus here is to investigate ways in which this choice can be made in the most objective way possible. A growing field of work on deriving objective prior distributions has emerged in the last few decades, mainly pioneered by Harold Jeffreys. These are referred to as *non-informative priors* and are discussed in detail in Section 2.2.4.

2.2.2 Loss functions and Bayesian estimators

2.2.2.1 Risk and estimation in the Bayesian setting

Clearly, one of the most attractive aspects of Bayesian statistics is that the posterior distribution of the parameter on which inference is performed, provides a very wide scope for analysis. However, since one is left with much more than a mere point estimate, some general way is needed to derive estimators that work in an optimal manner with regards to the posterior.

In order to develop the idea of Bayesian estimators, concepts of loss and risk first need to be defined. The discussions in the remainder of this chapter regard a general posterior

distribution $\pi(\theta|\mathbf{t})$; specific estimators relating to the compound Rayleigh distributions will be derived in Chapter 3. Denote the expected value under the posterior distribution as $\mathrm{E}_{\theta|\mathbf{t}}[\theta]$. Consider now an arbitrary estimator $\hat{\theta}$ of the parameter, as well as a loss function $L(\theta, \hat{\theta})$. Loss functions will be described in the following section, but in broad terms L portrays the nature of the loss incurred between the true parameter value and an estimate of that parameter.

With this in mind, the *Bayes risk* is now defined as the posterior expected loss, that is,

$$\mathrm{E}_{\theta|\mathbf{t}}[L(\theta, \hat{\theta})] = \int_{\Theta} L(\theta, \hat{\theta})\pi(\theta|\mathbf{t})d\theta. \tag{2.5}$$

Finally, an estimate $\hat{\theta}$ is referred to as a Bayesian estimator if it minimises the Bayes risk, such that

$$\hat{\theta}_{\text{Bayes}} = \arg\min_{\hat{\theta}} \mathrm{E}_{\theta|\mathbf{t}}[L(\theta, \hat{\theta})].$$

Thus, a procedure relying on the intuitive concept of risk (or loss) minimisation is available for automatically deriving an estimator from any posterior distribution. Bayesian estimators can also be shown to have good asymptotic properties, such as consistency (converges to the true value almost surely) and relatively efficient convergence to the true value (Robert, 2007).

2.2.2.2 Quantifying loss in estimation

Loss functions are commonly found across many mathematical fields dealing with optimisation or modelling. Sometimes, problems are stated in terms of the negative of the loss function, which is called the utility function, but the idea remains the same. In all paradigms dealing with statistical estimation, loss functions play an integral role in providing a way to tell the analyst when an optimal value has been reached.

Essentially, such a function is a mapping between different values of a parameter (or parameters) to a non-negative real number that signifies the nature of the cost or loss incurred, usually between the true value of the parameter and its estimate. The most trivial example of a loss function is the 0-1 loss, which uses the indicator function,

$$L_{\text{0-1}}(\theta, \hat{\theta}) = \begin{cases} 0 & \text{if } \theta = \hat{\theta} \\ 1 & \text{if } \theta \neq \hat{\theta} \end{cases} \tag{2.6}$$

such that when the estimate differs from the true value, a loss of 1 is measured, otherwise it is 0.

Determining the nature of loss for a specific problem is not trivial. To truly specify the correct function, one would need knowledge of the extent of the loss incurred as the estimation error changes, but this can only happen in a theoretical context – in practice estimation error cannot be known exactly. As a result, loss functions are often chosen for their convenient mathematical form. Symmetric loss functions are usually assumed, but in many problems under- and overestimation of the parameter should not accrue the same penalty.

In this thesis, an attempt is made to showcase and compare the results obtained over a range of loss functions. Four versions of each estimator are derived, corresponding to two symmetric and two asymmetric loss functions.

One of the most intuitive loss functions considers only the absolute difference between θ and its estimate $\hat{\theta}$, since a negative loss has no meaning. Accordingly, it is referred to as absolute error (AE) loss and has the form

$$L_{\mathrm{AE}}(\theta, \hat{\theta}) = |\theta - \hat{\theta}|. \tag{2.7}$$

Laplace considered this loss function when formally deriving the first Bayesian estimator while working on problems in astronomy (Stigler, 1986). The drawback of AE loss is that the absolute value operator complicates further mathematical manipulations, e.g. it is not differentiable where $L_{\mathrm{AE}}(\cdot) = 0$.

Squared error (SE) loss is nearly ubiquitous in statistical modelling and presents a mathematically tractable alternative to AE loss. Its simplicity and relatively intuitive nature usually trumps objections against its arbitrary use. It is defined as

$$L_{\mathrm{SE}}(\theta, \hat{\theta}) = (\theta - \hat{\theta})^2. \tag{2.8}$$

As the discrepancy between a true value and its estimate increases, the loss grows quadratically, instead of linearly as with L_{AE} in (2.7). Thus, outlying values can be responsible for the largest part of the total lost incurred, leading to a skewed picture of the average loss across a range of estimates. Even though some work has been done to address this, such as the Huber loss, a piecewise loss functions which carries a linear

penalty after a certain estimation error cut-off value, only AE and SE loss are considered here due to their generality in form.

In addition, two asymmetric loss functions are used, namely linear exponential (LINEX) loss and general entropy (GE) loss. The forms of both of these loss functions are controlled by an additional hyper-parameter, allowing the analyst to adjust the form of the loss depending on the direction and degree of asymmetrical penalisation.

LINEX loss, with its parameter a, can be defined as (Zellner, 1986)

$$L_{\text{LNX}}(\theta, \hat{\theta}) = e^{a(\hat{\theta}-\theta)} - a(\hat{\theta} - \theta) - 1, \tag{2.9}$$

and using a series expansion to represent e, it can easily be seen that LINEX loss approximates a symmetric SE loss as a tends toward 0 from either side. It is known that

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \qquad \forall x$$

thus, for small values of $|a|$, it follows that

$$\begin{aligned} L_{\text{LNX}}(\theta, \hat{\theta}) &= \left(1 + a(\hat{\theta} - \theta) + \frac{a^2}{2}(\hat{\theta} - \theta)^2 + \dots\right) - a(\hat{\theta} - \theta) - 1 \\ &= \frac{a^2}{2}(\hat{\theta} - \theta)^2 + \dots \\ &\approx \frac{a^2}{2}(\hat{\theta} - \theta)^2, \end{aligned}$$

such that L_{LNX} is predominantly influenced by a symmetric squared term. The magnitude of the parameter a controls the extent of the asymmetry, while its sign determines the direction. Moreover, positive values of a will lead to a greater cost for overestimation, while negative values of a will penalise underestimation more.

General entropy loss is similar in nature to LINEX loss, with an accompanying parameter k that controls the degree and direction of asymmetry. It is defined to be

$$L_{\text{GE}}(\theta, \hat{\theta}) = \left(\frac{\hat{\theta}}{\theta}\right)^k - k \ln\left(\frac{\hat{\theta}}{\theta}\right) - 1. \tag{2.10}$$

Guure et al. (2012) use the GE loss function in comparison with LINEX and SE loss to derive Bayesian estimators for a Weibull model in a survival analysis context. Similar to

the parameter of LINEX loss, positive values of the hyper-parameter k lead to a greater penalisation for overestimation and *vice versa*.

The loss functions described above are presented visually in Figure 2.1. Notice how the asymmetry of LINEX and GE loss changes with different values of their respective hyper-parameters. It is also interesting to note that for LINEX loss, the resulting curve for a and $-a$ is mirror-imaged. The same is not true in the case of GE loss.

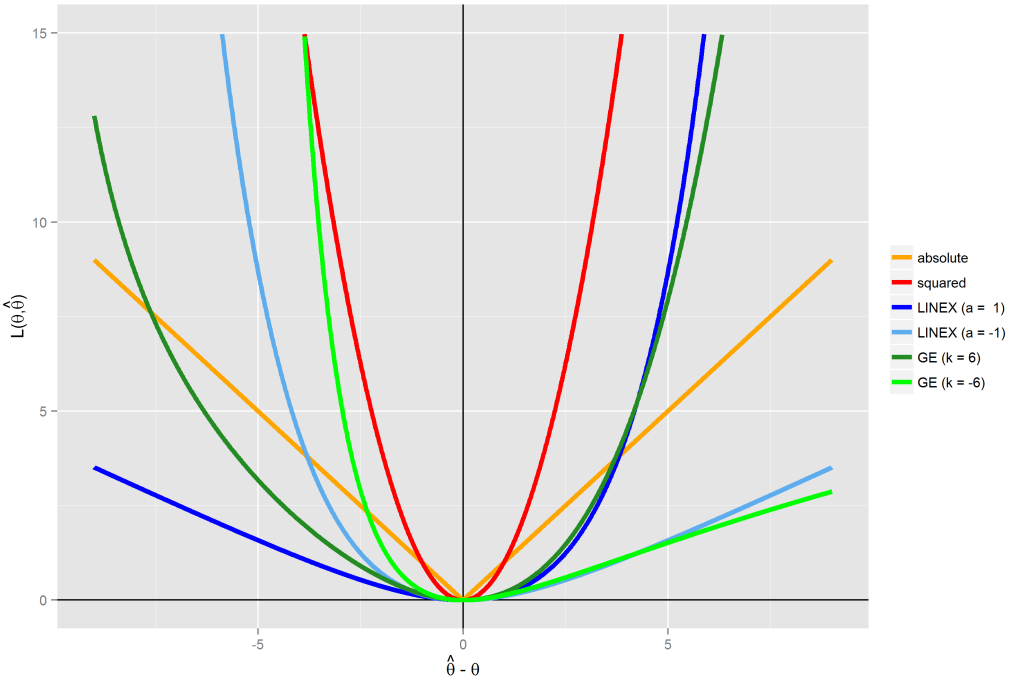


Figure 2.1: *The loss functions used in this thesis depicted visually against the difference between true parameter value and arbitray estimate. These include the symmetric AE and SE loss functions and asymmetric LINEX and GE loss functions.*

2.2.2.3 Derivations of Bayesian estimators

Following the specifications of loss functions that will be used, general Bayesian estimators can now be derived in terms of an arbitrary posterior distribution $\pi(\theta|\mathbf{t})$. Recall that a Bayesian estimator minimises the Bayes risk (2.5). For a given loss function, this is done in the usual manner by simplifying the integral, taking the derivative with respect to θ and equating to 0.

Table 2.1 below summarises the derivations of the Bayesian estimators that are used in the current work. The notation $\text{median}_{\theta|\mathbf{t}}[\theta]$ denotes the median of the posterior distribution. This can formally be defined as the lowest possible value θ^* for which

$$P(\theta \geq \theta^*) \geq \frac{1}{2}$$

is satisfied.

Table 2.1: *Bayesian estimators for a general posterior distribution for various loss functions.*

name	loss function	Bayesian estimator
0-1 loss	(2.6)	$\hat{\theta}_{\text{0-1}} = \text{mode}_{\theta\|\mathbf{t}}[\theta]$
AE loss	(2.7)	$\hat{\theta}_{\text{SE}} = \text{median}_{\theta\|\mathbf{t}}[\theta]$
SE loss	(2.8)	$\hat{\theta}_{\text{AE}} = \text{E}_{\theta\|\mathbf{t}}[\theta]$
LINEX loss	(2.9)	$\hat{\theta}_{\text{LNX(a)}} = -\frac{1}{a}\ln \text{E}_{\theta\|\mathbf{t}}[e^{-a\theta}]$
GE loss	(2.10)	$\hat{\theta}_{\text{GE(k)}} = \left(\text{E}_{\theta\|\mathbf{t}}[\theta^{-k}]\right)^{-\frac{1}{k}}$

One can see that estimators corresponding to the two symmetric loss functions, AE and SE loss, are reduced to the posterior median and posterior mean respectively, both of which are popular and sensible measures. LINEX and GE loss lead to slightly more complicated functions involving the posterior expectation. Interestingly, it is clear that with a symmetry parameter value of $k = -1$, the Bayesian estimator for GE loss is equivalent to $\hat{\theta}_{\text{SE}}$. It is important to note that while these derivations are for estimators of the parameter, they can be extended to functions of the parameter θ without loss of generality. Thus, as an example, the Bayesian estimator for any function of the parameter $m(\theta)$ under SE loss will also be the posterior mean of $m(\theta)$.

2.2.3 The Fisher information

2.2.3.1 Background and relevance to current work

Fisher information, an important statistical concept, is applicable to Bayesian methods as well as other types of estimation theory such as maximum likelihood. Depending on the context, the word "information" can take on different exact meanings and it can be elusive to define rigorously. Claude Shannon developed an entire field now referred to as Information Theory out of work he did on signal processing (Shannon, 1948). Here, the idea of entropy plays an important role – this can loosely be defined as a measure of uncertainty pertaining to the prediction of a random variable's value. This uncertainty can be conceptualised as the inverse of information.

At a fundamental level, information can be thought of as the propagation of cause and effect within a system. It can then become a measure of how well the state of one part of the system can be known through observation in another part of the system. In the early 20th century, the esteemed biologist and statistician Ronald Fisher started to develop a definition of information in the context of statistical estimation (Fisher, 1925). Since then, Fisher information has been widely studied and applied. One can think of the Fisher information as a measure of information that a random variable (and its observed data) provides about an unknown parameter in a probabilistic model.

Suppose an arbitrary probability model f for a random variable T has parameter(s) θ. The Fisher information $\mathcal{I}_F$ is then defined as the expected value of the partial derivative with respect to θ, such that

$$\mathcal{I}_F(\theta) = \mathrm{E}_T\left[\left(\frac{\partial \ln f(t|\theta)}{\partial\theta}\right)^2\right] = -\mathrm{E}_T\left[\frac{\partial^2 \ln f(t|\theta)}{\partial\theta^2}\right] \tag{2.11}$$

providing the second derivative exists.

If the model has multiple parameters, i.e. $\theta = (\theta_1, \theta_2, \ldots, \theta_p)$, the Fisher information is a symmetric and positive semi-definite $p \times p$ matrix with element in row i and column j defined as

$$[\mathcal{I}_F(\theta)]_{i,j} = -\mathrm{E}_T\left[\frac{\partial^2 \ln f(t|\theta)}{\partial\theta_i \partial\theta_j}\right].$$

In asymptotic maximum likelihood theory, the Fisher information is equivalent to the inverse of the covariance matrix. This emphasises that in order to increase precision, minimising the variance is akin to maximising the information. In fact, the Fisher information can be used to derive a lower bound on the variance of an unbiased estimator. This is known as the Cramer-Rao bound or information inequality (Wackerley et al., 2008).

In the realm of Bayesian statistics, the Fisher information can be very useful. Even though we do not usually care about asymptotic behaviour, a very important and reassuring result, the Bernstein-von Mises theorem, states that given enough data, the posterior distribution is not dependent on the prior, only the Fisher information (Rivoirard and Rousseau, 2012). Additionally, the Fisher information is also an important factor in the derivation of a number of non-informative prior distributions, including those discussed in Section 2.2.4.

2.2.3.2 Numerical approximation of the Fisher information

The derivation and calculation of the Fisher information is very important in this study specifically. It is used to obtain the forms of the non-informative prior distributions of interest and consequently also plays a critical role in the process of simulating realisations from the posteriors. However, due to the mathematical nature of the generalised compound Rayleigh models, closed form solutions of the Fisher information cannot be obtained. Therefore, ways of estimating or approximating the value of the Fisher information matrices for given parameter values need to be considered.

Das et al. (2010) employ a resampling scheme to obtain an estimator with good statistical qualities of the Fisher information matrix in settings where its derivation lead to analytical difficulties. Their method can incorporate prior information and make use of perturbed versions of the data, but is quite elaborate.

The technique followed in this thesis, numerical integration or quadrature, is more direct. In their seminal text, Piessens et al. (1983) not only provides an overview of the available numerical integration routines, but also programmed these collection of algorithms in a standard, efficient and functional way. Their package of automatic integration software has become the industry standard and is implemented in most scientific computing

platforms today. It is automatic in the sense that these algorithms can be called with only such inputs as the integrand, integration bounds and accuracy tolerances. The specific routine employed here is called adaptive Gauss-Kronrod quadrature, with extrapolation by Wynn's ϵ-algorithm.

A detailed overview of adaptive quadrature techniques constitutes a field of study on its own and is beyond the scope of the current work. For this reason, only a brief introduction to the fundamental principles are discussed here, based on the discourse of Piessens et al. (1983).

In all its forms, numerical quadrature techniques approximates the solution of the integral

$$I = \int_a^b f(x)dx$$

in light of two measures of tolerance, the absolute accuracy ϵ_{a} and the relative accuracy ϵ_{r}. In its simplest form, quadrature methods approximate I by evaluating the integrand $f(x)$ at various points between the bounds a and b in a linear or non-linear manner. Adaptive quadrature methods evaluate the integrand in such a way that more points are chosen at the regions where evaluation is difficult, a process dictated by an error estimate which sequentially updates. Infinite bounds can be handled in a number of ways, such as mapping to a finite interval.

Accordingly, adaptive quadrature is a stepwise procedure and at each step i, one obtains an approximation to I based on n_i function values, denoted R_{n_i}, as well as an error estimate E_{n_i}. In the event that convergence can be attained, the procedure is terminated once the condition

$$|R_{n_i} - I| \leq E_{n_i} \leq \max\left(\epsilon_{\mathrm{a}}, \epsilon_{\mathrm{r}}\right)$$

is satisfied. The adaptive nature means that the number of function evaluations n_i is determined during the course of the algorithmic execution, rather than being predetermined. The approximations to I are quantified in terms of quadrature sums, defined as

$$Q_n(a,b)f = \sum_{k=1}^{n} w_k f(x_k) \approx \int_a^b w(x)f(x)dx,$$

where the weights $w_1, w_2, \ldots w_n$ correspond to the points of evaluation $x_1, x_2, \ldots x_n$, called abscissae or nodes. Different quadrature methods have different ways of assigning a and b and the weight function $w(x)$. Gauss-Legendre quadrature, the method

considered here, considers $a = -1$, $b = 1$ and $w(x) = 1$. An important adjustment by Kronrod (1965) to the number of abscissae considered results in a more efficient way to simultaneously calculate the integral approximation and its error at each step.

An analysis of the error term can lead to limiting procedures for improved accuracy of approximation, such as adjusting the sub-intervals of integration. In the Gauss-Krondrod adaptive quadrature routine, this is facilitated using the recursive ϵ-algorithm of Wynn (1956).

In closing, adaptive quadrature allows the numerical evaluation of the value of some elements of the Fisher information matrix when an analytical solution is not available. This method is both straightforward and computationally efficient.

2.2.4 Non-informative approach to prior distribution specification

The choice of prior distribution is central to the Bayesian method, but also quite contentious. When previous knowledge of the data or experiment being analysed is available, it can be beneficial to include this in the form of a prior. This presents a useful way to build on the results of previous similar studies or domain knowledge. However, it is also one of the drawbacks, since one can easily argue about the integrity of this previous knowledge, as well as the exact way in which its incorporation takes place. In cases where no previous information is available, a Bayesian analysis still requires prior specification. For these reasons, it was critical for methods to be developed that strives to derive prior distributions in the most objective manner possible, leading to so-called non-informative priors.

Non-informative priors do suffer from some drawbacks. They are usually also improper, meaning that they cannot be normalised to integrate to unity as a probability distribution should. Even so, this is allowable in Bayesian statistics, as long as the resulting posterior distribution is well-defined. Moreover, use of non-informative priors may result in the violation of the likelihood principle (Berger and Bernardo, 1992). Thus, when performing analyses, it is advisable to do sensitivity tests by comparing the output of different non-informative priors.

Three types of non-informative prior distribution are considered in the current work, all of which are derived from the Fisher information. These are the Jeffreys prior, the

reference prior and the probability matching (PM) prior. In the remainder of this section, a brief overview regarding the motivation for and derivation of each will be presented.

2.2.4.1 The Jeffreys prior

Historically, Laplace used the uniform distribution as a prior for demonstration in cases where no assumptions were made with regards to prior knowledge. The major criticism that partly lead to the eventual decline in the use of Bayesian statistics was that this prior would change with a transformation of the parameters (Berger and Bernardo, 1992).

Jeffreys (1946) was one the first to address this problem by devising a prior distribution that is invariant under reparameterisation. As a result, it is only dependent on the model chosen for the data. It is derived by simply taking the square root of the Fisher information in the univariate case, or the square root of its determinant in the multivariate case. Thus, we have

$$\pi_{\text{jeff}}(\theta) = \sqrt{\mathcal{I}_F} \qquad \text{(univariate case)}$$
$$\pi_{\text{jeff}}(\theta) = \sqrt{|\mathcal{I}_F|} \qquad \text{(multivariate case)}$$

where θ represents the arbitrary model parameter(s) and $\mathcal{I}_F$ is the Fisher information.

2.2.4.2 The reference prior

The reference prior was developed as a solution for the Bayesian analyst who wishes to stay objective without regard to the situation in which it is used. While Jeffreys succeeded in setting out a method to obtain a non-informative prior with the property of invariance, his prior is primarily suited to cases in which the parameter of interest is univariate. The reference prior method, in some sense, extends the work done by Jeffreys and produces a non-informative prior with attractive properties even when model parameters are plentiful. Even though it is unlikely that there will ever be an industry standard non-informative prior, the reference prior will be a prominent contender to fill such a role, due to its wide applicability. In the remainder of this section, a brief overview of steps involved in the derivation of a reference prior will be given, largely based on

the discussion of Berger and Bernardo (1992), where the pioneers of the method provide a summary of the motivation, implementation and other technical aspects of reference priors since its inception more than a decade earlier (Bernardo, 1979).

The reference prior is particularly appealing because it is formulated to contain the least amount of information with regard to the parametric model. This is achieved by basing the prior's derivation on the Kullback-Leibler (KL) divergence. In essence, the KL divergence is an attempt to quantify the distance (or *divergence*) between two statistical densities. This allows us to specify a non-informative prior in a very intuitive manner.

Denote the KL divergence between two density functions $f(\theta)$ and $g(\theta)$ as $D(f, g)$. It can be defined as

$$D(f, g) = \int_{\Theta} f(\theta) \ln \left(\frac{f(\theta)}{g(\theta)} \right) d\theta.$$

The KL divergence is not strictly a distance metric, since it is asymmetric: in general $D(f, g) \neq D(g, f)$. It is more accurate to state that the KL divergence measures the loss of information incurred by approximating the density f by g.

Consider now a random variable T and a corresponding set of iid observations, denoted as usual by $\mathbf{t}$, and a model parameter θ for which a prior distribution $\pi(\theta)$ is sought. A natural choice for a non-informative prior now emerges as one which maximises the expected divergence between the prior and consequent posterior,

$$\mathrm{E}_T[D\{\pi(\theta|\mathbf{t}), \pi(\theta)\}], \tag{2.12}$$

where the expectation is over the marginal density of the data. However, the resulting prior distribution is usually discrete and therefore impractical (Berger and Bernardo, 1992). While introducing the reference prior, Bernardo (1979) tried to alleviate this problem with (2.12) by considering that the data has n elements, $\mathbf{t} = (t_1, t_2, \ldots, t_n)$, and changing the maximisation problem such that

$$\lim_{n \to \infty} \mathrm{E}_T[D\{\pi(\theta|\mathbf{t}), \pi(\theta)\}] \tag{2.13}$$

yields the prior distribution. The idea behind this was that as n grows large enough, the data will provide perfect information about the parameter θ, such that (2.13) amounts to the information not known about θ through the prior $\pi(\theta)$. The maximisation thus

yields the least informative prior. Regrettably, this maximum is usually infinite, but one can modify the method to rather find a prior using the expected divergence above for each n, and then obtaining the reference prior as the limit of those priors as n tends towards infinity (Berger and Bernardo, 1992).

Generally, the process of deriving a reference prior starts by allocating model parameters in groups according to their inferential importance, with the parameter(s) of interest first. As a result, the reference prior method produces different priors for any given model, depending on the ordering used. Berger and Bernardo (1992) advise to use the one-at-a-time rule, where the number of groups is equal to the number of parameters. If no clear distinction between the interest of model parameters exist, they also advise to derive a reference prior for each distinct ordering if possible. Then, the analysis can continue with a multitude of reference priors which can be compared against each other. This is the practice followed in this thesis.

Finally, the steps involved for obtaining the reference prior in a simplified setting is briefly discussed, where $\theta = (\theta_1, \theta_2)$, with θ_1 the parameter of interest. The notation set out in Berger and Bernardo (1989) is used. The process starts by choosing a conditional prior distribution $\pi(\theta_2|\theta_1)$. A sensible choice here is Jeffreys' prior, such that

$$\pi(\theta_2|\theta_1) \propto \sqrt{[\mathcal{I}_F(\theta)]_{22}},$$

making use of the bottom right element of the 2×2 Fisher information matrix.

Next, with Θ denoting the parameter space of θ, a sequence of subsets $\Theta_1 \subset \Theta_2 \subset \ldots$ is chosen such that their union is the parameter space: $\bigcup_i \Theta_i = \Theta$ and the density $\pi(\theta_2|\theta_1)$ is not infinite on the subspace $\Omega_{i,\theta_1} = \{\theta_2 : \theta \in \Theta_i\}$ for all θ_1. Now, define $K_i(\theta_1) = \left(\int_{\Omega_{i,\theta_1}} \pi(\theta_2|\theta_1) d\theta_2\right)^{-1}$ in order to obtain normalised conditional densities

$$\pi_i(\theta_2|\theta_1) = K_i(\theta_1)\pi(\theta_2|\theta_1), \tag{2.14}$$

for each parameter subspace Ω_{i,θ_1}. This allows the computation of the marginal prior of θ_1,

$$\pi_i(\theta_1) = \exp\left\{\frac{1}{2}\int_{\Omega_{i,\theta_1}} \pi_i(\theta_2|\theta_1) \ln\left(\frac{|\mathcal{I}_F(\theta)|}{[\mathcal{I}_F(\theta)]_{22}}\right) d\theta_2\right\}. \tag{2.15}$$

In conclusion, the joint prior of θ_1 and θ_2, the reference prior, is now defined as the limit over the parameter space subsets,

$$\pi(\theta) = \lim_{i\to\infty} \frac{K_i(\theta_1)\pi_i(\theta_1)}{K_i(\theta^*)\pi_i(\theta^*)}\pi(\theta_2|\theta_1), \tag{2.16}$$

where θ^* is any fixed value of θ_1. A second reference prior can now be obtained by repeating this process, but first choosing a conditional prior $\pi(\theta_1|\theta_2)$.

Interestingly enough, it can be shown that the reference prior is equivalent to the Jeffreys prior if the posterior distribution is asymptotically normal (Bernardo, 1979). For a more detailed description of the derivation process, the reader is directed to Berger and Bernardo (1992).

2.2.4.3 The probability matching prior

The last type of prior under consideration arises from an interesting question. The PM prior strives to show how one can construct a prior distribution where both frequentist and Bayesian probabilities coincide up to a certain degree of error. It should be noted that the nature of the parameter θ is fundamentally different in these two paradigms, since the former considers it deterministic and the latter assigns to it a probability distribution, casting the parameter as a random variable. Nevertheless, frequentist theory is extensive and powerful in many settings, and specifying a prior with similar properties may be a worthwhile endeavour in an objective Bayesian analysis.

The derivation of the PM prior was done by Datta and Ghosh (1995), and it is their formulation that is discussed here. Consider, as usual, a parametric model $f(t|\theta)$ with p parameters, $\theta = (\theta_1, \theta_2, \ldots, \theta_p)$. Furthermore, suppose a sample $\mathbf{t}$ of n iid realisations from this model is available, as well as a real-valued twice continuously differentiable function of the parameters $a(\theta)$. Let $\hat{\theta}$ represent the posterior mode or maximum likelihood estimate, and b the (asymptotic) posterior variance of $\sqrt{n}\{a(\theta) - a(\hat{\theta})\}$.

For all z, the PM prior satisfies the condition

$$P_\theta\left(\frac{\sqrt{n}\{a(\theta) - a(\hat{\theta})\}}{\sqrt{b}} \leq z\right) = P_{\pi(\theta|\mathbf{t})}\left(\frac{\sqrt{n}\{a(\theta) - a(\hat{\theta})\}}{\sqrt{b}} \leq z\right) + \mathcal{O}_p(n^{-1}). \tag{2.17}$$

Thus, for all z, the probabilities given by the frequentist confidence interval and the Bayesian credible interval under the posterior distribution is equivalent. Datta and Ghosh (1995) show that this equality is achieved if and only if

$$\sum_i \frac{\partial}{\partial \theta_i}\{\eta_i(\theta)\pi(\theta)\} = 0, \tag{2.18}$$

where the vector $\eta(\theta) = (\eta_1(\theta), \eta_2(\theta), \ldots, \eta_p(\theta))^T$ is given by

$$\eta(\theta) = \frac{\mathcal{I}_F^{-1}\nabla_a(\theta)}{\sqrt{\nabla_a^T(\theta)\mathcal{I}_F^{-1}\nabla_a(\theta)}}.$$

In the above, $\mathcal{I}_F^{-1}$ is the inverse of the Fisher information matrix, and $\nabla_a(\theta)$ is the vector of first order differentiates, i.e.

$$\nabla_a(\theta) = \left(\frac{\partial a(\theta)}{\partial \theta_1}, \frac{\partial a(\theta)}{\partial \theta_2}, \ldots, \frac{\partial a(\theta)}{\partial \theta_p}\right)^T.$$

By solving the first order differential equation (2.18) for π, the PM prior that matches frequentist and Bayesian probabilities as in (2.17) is obtained.

2.2.5 Simulating the posterior distribution

Simulation is a very important tool, providing researchers in all fields of science with an invaluable way of testing theories and models. In recent years, exponential growth in computational processing power has meant that simulation studies can now be employed on large scales. In Statistics especially, simulation has emerged as an indispensable instrument. Although simulated data cannot replace actual real-world data, it provides an incredibly convenient way to conduct statistical research and test models under predefined assumptions.

In this thesis, simulation plays a central underlying role in the study of the compound and generalised Rayleigh models and the comparison of the different elements used in the Bayesian estimation procedure. Initially, survival data samples are generated from the Rayleigh models' distributions for assessment purposes. However, the power of simulation really comes into play by allowing us to generate realisations that are approximately distributed according to the model parameters' posterior distributions,

which cannot be obtained analytically, and in doing so, estimate properties of these posteriors. This is done by using a MCMC method, which will be discussed in Section 2.2.5.1. The chapter concludes with a note on the convergence of the simulations.

2.2.5.1 MCMC and the Metropolis-Hastings algorithm

MCMC encompasses a class of techniques that is well-established, vastly applied and the subject of innumerable books and papers in the literature. A detailed study of the background, formulation and underlying theory is beyond the scope of this thesis, since it is only employed as a means to an end. In this section, only a brief description of MCMC and the algorithm used for its implementation, referred to as *Metropolis-Hastings*, is provided. See, for example, Gilks et al. (1996) for a much more elaborate text on the practical implementation of MCMC methods.

The two "MCs" in MCMC signifies the unification of two powerful concepts. Monte Carlo methods involve the generation (or simulation) of random values from some probability mechanism, such as a distribution. Markov theory is concerned with systems or processes where a transition between states occurs with concurrent steps in time. A chain of values from such a process is then called a Markov chain if it satisfies the Markov property. This property specifies a certain lack of memory, whereby the future state of the chain is only dependent on the current state. Mathematically, the chain emerges as random variables $X_1, X_2, X_3, \ldots$ defined in a state space, for which

$$\mathrm{P}(X_{t+1} = x | X_1 = x_1, X_2 = x_2, \ldots, X_t = x_t) = \mathrm{P}(X_{t+1} = x | X_t = x_t)$$

holds. Accordingly, after sufficient steps and given certain regularity conditions, a Markov chain ignores the initial states it started from and converges to the so-called *stationary distribution*, where transition probabilities between states are fixed. This stationary distribution is the main outcome of MCMC methods and provides the probability mechanism that drives the simulation.

The remainder of this discussion is loosely based on the discourse of Rizzo (2008). MCMC simulation is often used to approximate integrals that are very complex or mathematically intractable. Another popular application is the sampling from distributions for which a closed-form solution cannot be derived analytically. This happens frequently

in Bayesian analyses, and is also the case with the posterior distributions of the model parameters in this thesis. One refers to the desired distribution from which samples are to be generated as the target distribution, for which the normalising constant is unknown.

The most common way of implementing the MCMC procedure described above, is with the Metropolis-Hastings algorithm. The problem was first postulated and formalised by Metropolis et al. (1953) and later generalised by Hastings (1970). A simplified version of the algorithmic steps are summarised in Figure 2.2 below, followed by a brief explanation. A general target distribution f (which needs to be known only in proportional form) is considered, as well as a proposal distribution, g, from which potential samples are generated as candidates.

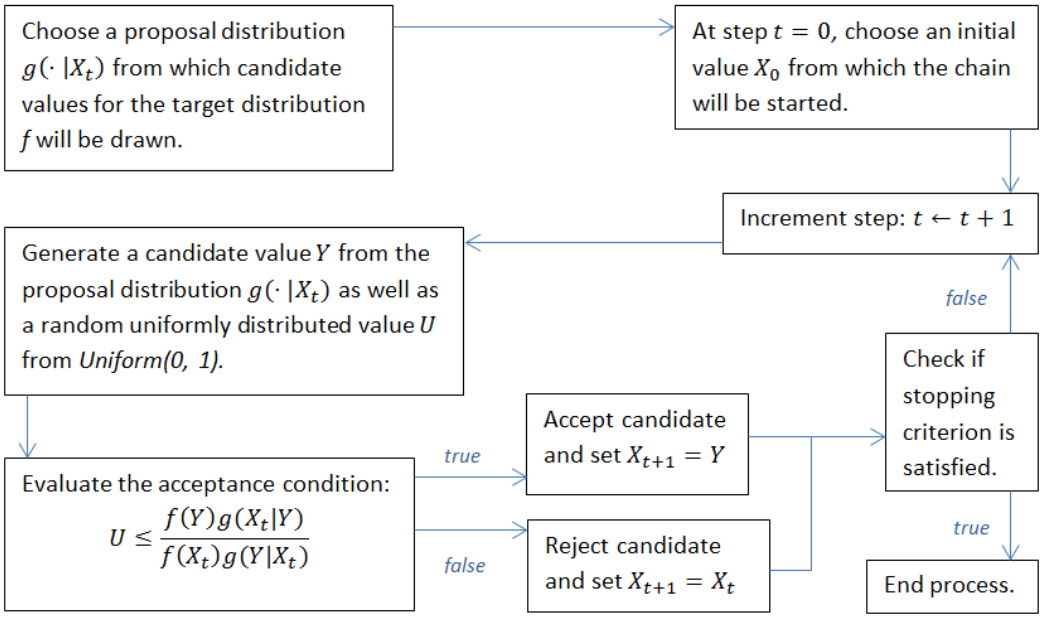


Figure 2.2: *The general form of the Metropolis-Hastings algorithm summarised as a flowchart.*

As Figure 2.2 shows, realisations are drawn from a proposal distribution g, acting as candidate values for the target f. An acceptance condition relying only on the previously accepted value is evaluated for a new candidate, otherwise the state of the chain remains at the previous value. The proposal distribution should ideally resemble the target distribution from which samples are desired, but it is somewhat arbitrary. One of the remarkable aspects of the Metropolis-Hastings algorithm is that under mild regularity conditions, the choice of proposal distribution does not affect the eventual outcome of obtaining samples from f. It does, however, affect the rate of convergence and the

efficiency of the generated chain. The problem of monitoring convergence is discussed in Section 2.2.5.2.

Furthermore, the algorithm can be extended for a multivariate target distribution in a number of ways. The approach followed here is referred to as *blocking*. Suppose one wants to generate realisations from a target density f with j dimensions (parameters). For each step t the outcome is $X_t = (X_{t1}, X_{t2}, \ldots, X_{tj})$. Now, at a given step, each of the dimensions is updated with the acceptance or rejection of an independent candidate point incrementally, each time using the most up-to-date chain values. For example, if the first two dimensions have already been updated and one wishes to evaluate the acceptance condition for the third dimension, the form of the target density used would be $f(\cdot|X_{(t+1)1}, X_{(t+1)2}, X_{t3}, \ldots, X_{tj})$.

After an initial number of chain values has been generated, the *burn-in period*, convergence is assumed to have taken place and all new samples drawn will be approximately distributed according to the stationary distribution. Thus, by choosing the posterior distributions, which are only known up to the normalising constant, as the target distribution, a large number of samples can be acquired. From these, characteristics such as the mean, median, variance and percentiles of the distribution can be estimated. The algorithm is usually terminated after the chain has reached a suitable length to accurately calculate these estimates.

2.2.5.2 Monitoring convergence of Markov chains

MCMC simulation is described in the previous section as a technique which can be used to draw random observations from complex distributions which is only known in its proportional form, by producing a chain of values which ideally converges to the distribution of interest. The ergodic theorem, a standard Markov theory result, ensures that the simulated chain will eventually become the desired target distribution, however, a huge concern is that there is no clear indication of when this convergence will take place.

The most prominent problems faced when monitoring convergence, in comparison with other optimisation algorithms, is firstly due to the stochastic nature of MCMC chains, meaning that one cannot rely on the monotonicity of the algorithm. Secondly, the

convergence of a distribution is monitored, not a single point (Gilks et al., 1996). Some methods have been developed to overcome these problems and the technique devised by Gelman and Rubin (1992) will be considered here.

Gelman and Rubin (1992) argue that convergence cannot be established based on a single simulated chain. This is sensible, since a MCMC may get stuck in certain regions for long periods of time, conveying a false notion of convergence. This may be due to inherent slow evolution of the chain, but can also be influenced by the initial value of the MCMC algorithm. The ergodic theorem states that the generated chain will eventually "forget" its initial value, but since the task is to determine how many iterations this will take, it makes sense to monitor multiple chains with distinct initial values simultaneously. Ideally, these initial values should be overdispersed across the domain of the target distribution. Gelman and Rubin (1992) discuss in great detail a procedure to obtain approximations of the mode(s) of the target distribution and thereby constructing an approximate mixture distribution from which overdispersed values could be drawn. However, at the time the paper was published, computational power was a big aspect, and this procedure was followed to speed up the convergence of the algorithm. Nowadays, constructing MCMC chains with tens of thousands of iterations takes an almost insignificant amount of computational time, thus it is deemed that selecting initial values overdispersed in the target distribution can be done much more informally – with enough time the chain will drift away from these starting points towards the target distribution in any case.

A simple way to gauge whether convergence has taken place based on multiple chains may involve plotting each of the chains on the same set of axes and determining when they are overlayed to such an extent that they are indistinguishable. The method described here is based on this concept, but achieves this by using calculations regarding components of variance instead of graphical inspection. The method, as described in Gilks et al. (1996), follows.

The monitoring of the convergence of the distribution amounts to monitoring of the convergence of all scalar summaries of interest, such as the parameters of the the distribution or summary statistics of interest. Let ψ_{ij} be a general scalar summary of interest, the jth value of the ith chain, in a scenario where m MCMC sequences are investigated, each of length n, started from distinct initial values. Interest lies in both the variance

within each chain and between the different chains. Thus, consider the variance between the m chain means

$$\frac{B}{n} = \frac{1}{m-1}\sum_{i=1}^{m}\left(\overline{\psi}_{i\cdot} - \overline{\psi}_{\cdot\cdot}\right)^2$$

where

$$\overline{\psi}_{i\cdot} = \frac{1}{n}\sum_{j=1}^{n}\psi_{ij} \qquad \text{and} \qquad \overline{\psi}_{\cdot\cdot} = \frac{1}{m}\sum_{i=1}^{m}\overline{\psi}_{i\cdot}$$

as well as the average of the variances within each chain

$$W = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n-1}\sum_{j=1}^{n}\left(\psi_{ij} - \overline{\psi}_{i\cdot}\right)^2.$$

An estimate of the variance of ψ, that is unbiased under the assumption that the initial values were drawn from the target distribution, can now be defined as

$$\widehat{\mathrm{V}}(\psi) = \frac{n-1}{n}W + \frac{1}{n}B.$$

Note that this estimate will be conservative (i.e. an overestimation), because the initial values were chosen to be overdispersed. It is also clear that the within-chain variance W underestimates the true variance of ψ, since there will be much less variability in a chain that has not yet covered all of the regions of the target distribution. Asymptotically, $\widehat{\mathrm{V}}(\psi)$ and W both approach the true variance of ψ, but from opposite directions, and this can now be used to get an idea of the possible scale reduction of the overestimated variance $\widehat{\mathrm{V}}(\psi)$. This factor is estimated as

$$\sqrt{\hat{R}} = \sqrt{\frac{\widehat{\mathrm{V}}(\psi)}{W}}.$$

As convergence is reached, this factor approaches 1, since the previously overestimated variance no longer needs to be reduced. A more statistically rigorous estimate of R can be derived (Gelman and Rubin, 1992), but the estimate presented here suffices in practice – $\hat{R}$ is used as a guideline for deciding after how many iterations convergence has been reached. Gilks et al. (1996) suggests to use a threshold value equal to 1.1 or 1.2. Finally, even though the formulation given here is in terms of a single summary of interest, convergence can be monitored across all parameters and scalar statistics of

interest by simply applying the procedure described here individually and ensuring that all values of $\hat{R}$ are close to 1.

2.3 Philosophical considerations

With any modelling endeavour, it is worthwhile to take a moment to reflect upon relevant choices made. This section is an attempt to briefly cast the problem statement of this thesis in a philosophical light. In broad terms, the simulation study performed here investigates Bayesian estimation of the parameters of a certain class of probability distributions, with an application in the field of survival analysis. Three aspects will be discussed: the choice of using parametric models, the choice of applying the Bayesian method and the choice of invoking the latter with objective prior distributions.

As Robert (2007) notes, Statistics should be seen as a tool for the interpretation of data, rather than its explanation. Any statistical inference, be it parametric or non-parametric in nature, is an attempt to use available data to draw conclusions about a larger population. The model or analytical framework preferably needs to mimic reality closely if it is to succeed in its task of providing a clear interpretation. However, the act of probabilistic modelling necessarily adds a layer of abstraction and will only ever be able to approximate reality to a certain extent. This does not obsolesce Statistics; in fact, it allows for efficient and often simplistic ways to gauge the extremely complicated (and unknowable) fabric of reality.

Generally, two probabilistic modelling approaches contend: parametric, where mathematical models are characterised by a set of explicitly defined parameters; and non-parametric, where more complex and adaptive probability structures are used to generate estimates. The latter makes fewer assumptions about the system that it models, and is able to produce very accurate results, even though its integrity is often based on asymptotic considerations. In spite of this, the parametric approach is opted for here. Assuming that a parametric model is the inherent driver of the system of interest is a hefty assumption, but when true, the superior modelling choice. Even if it only comes close, though, a parametric model allows for a much wider, simpler and more intuitive interpretation of a data set. The additional layer of simplification, from approximating

reality not only by a model, but by a parametric one, may also be advantageous in the sense of having a smoothing effect, i.e. to separate the signal from the noise.

Parametric models moreover allow for easy integration with Bayesian statistics, the inferential paradigm promoted here. Historically, many decades of controversy has not produced a victor between the Bayesian way and "classical" frequentist methods, and likely never will. Frequentist statistics is based on asymptotic formulations, and relies on large sample sizes for the accuracy of parameter estimates and their error bounds. The superiority of either of these perspectives are not debated, but rather some merits of the Bayesian approach are emphasised. First, conditioning on given data seems to be a much more intellectually appealing feature of an analysis than considering the hypothetical results of similar unobserved experiments (as required by frequentism). A second consideration is the important conceptual leap between the assumption of a model parameter that is unknown and deterministic, to the Bayesian viewpoint that it is random. The assignation of stochasticity to the parameter not only widens the inferential scope, but may also be closer to reality. We do not yet know the true nature of the universe in terms of its determinism, but even in cases where one can argue that a random parameter is senseless (such as an experiment that estimates a known physical constant), a probability distribution is the most useful tool for providing insightful output and quantification of error bounds. Therefore, one can argue the virtues of a prior for the parameter purely from a utilitarian perspective – the Bayesian approach does not necessitate the belief that the parameter is indeed bounded to a probability distribution.

The Bayesian method provides an intuitive, but also logically and axiomatically coherent way to update prior knowledge with data to arrive at a result. However, this prior information is overwhelmingly a subjective factor; its integrity could easily be a cause for concern and criticism. Philosophers from Emmanuel Kant to Karl Popper have mentioned the epistemological benefits of preconceived knowledge for scientific analyses, lest experiments be shrouded in sterility. Furthermore, it is inevitable that any statistical inference will have some level of subjectivity (even if it is only due to the analyst's choice of methodology). Nonetheless, we should strive for an approach that prioritises objectivity such that the data can predominantly speak for itself, especially in cases where prior knowledge is ill-determined. Non-informative priors, the type investigated in this thesis, allows for an objective Bayesian implementation.

A final consideration is that no single panacea, magic bullet algorithm, modelling approach or inferential paradigm exists that is enabled to encompass every statistical problem. One should not approach all analyses with a preconceived bias as to which method to apply, but rather strive to find an appropriate solution best suited to the problem at hand.

2.4 Summary

In this section, the most important results from this chapter are summarised.

In the context of survival analysis, a parametric approach is followed, whereby the most important parameters are the survival function,

$$S(t) = P(T > t) = 1 - F(t),$$

and especially the hazard rate,

$$h(t) = \frac{f(t)}{S(t)} = -\frac{\partial}{\partial t} \ln S(t) = \lim_{dt \to 0} \frac{P(t \leq T < t + dt | T > t)}{dt},$$

from equations (2.1) and (2.2).

The parametric approach implies the use of a probability model dependent on a set of parameters, denoted in general as $f(t|\theta)$. The corresponding likelihood function for a given data set $\mathbf{t}$, ordered such that $t_1 < t_2 < \ldots < t_d$ are events and $t_{d+1} < t_{d+2} < \ldots < t_n$ are right censored observations, is given in equation (2.3) as

$$\mathcal{L}(\theta|\mathbf{t}) = \frac{n!}{n-d} \prod_{i-1}^{d} f(t_i|\theta) \prod_{j=d+1}^{n} S(t_j)$$

to account for the partial information inherent in censored observations.

The estimation of parameter(s) θ follows the Bayesian paradigm, whereby a prior distribution $\pi(\theta)$ is updated to the posterior distribution with equation (2.4),

$$\pi(\theta|\mathbf{t}) = \frac{\mathcal{L}(\theta|\mathbf{t})\pi(\theta)}{\int_\Theta \mathcal{L}(\theta|\mathbf{t})\pi(\theta)d\theta}.$$

From the posterior, point estimates for the parameter, or functions thereof, can be obtained with Bayes estimators, defined as

$$\hat{\theta}_{\text{Bayes}} = \arg\min_{\hat{\theta}} \mathrm{E}_{\theta|\mathbf{t}}[L(\theta, \hat{\theta})],$$

where $L(\cdot)$ is one of four loss functions investigated in this thesis: absolute error, squared error, general entropy or linear exponential loss. The forms of these loss functions and their corresponding Bayesian estimators are summarised in the table below.

name	loss function	Bayesian estimator
AE loss	$\lvert\theta - \hat{\theta}\rvert$	$\hat{\theta}_{\text{AE}} = \text{median}_{\theta|\mathbf{t}}[\theta]$
SE loss	$(\theta - \hat{\theta})^2$	$\hat{\theta}_{\text{SE}} = \mathrm{E}_{\theta|\mathbf{t}}[\theta]$
LINEX loss	$e^{a(\hat{\theta}-\theta)} - a(\hat{\theta} - \theta) - 1$	$\hat{\theta}_{\text{LNX(a)}} = -\frac{1}{a}\ln \mathrm{E}_{\theta|\mathbf{t}}[e^{-a\theta}]$
GE loss	$\left(\frac{\hat{\theta}}{\theta}\right)^k - k\ln\left(\frac{\hat{\theta}}{\theta}\right) - 1$	$\hat{\theta}_{\text{GE(k)}} = \mathrm{E}_{\theta|\mathbf{t}}[\theta^{-k}]^{-\frac{1}{k}}$

The objective approach followed here necessitates the use of non-informative priors. The focus here will specifically be on Jeffreys' prior, the reference prior and the probability matching prior, for which motivations and derivation procedures are discussed in Section 2.2.4. These priors depend on the Fisher information, defined in general in equation (2.11) or element-wise for the multiple-parameter case, such that

$$[\mathcal{I}_F(\theta)]_{i,j} = -\mathrm{E}_T\left[\frac{\partial^2 \ln f(t|\theta)}{\partial\theta_i \partial\theta_j}\right],$$

if $\theta = (\theta_1, \theta_2, \ldots, \theta_p)$.

In the chapters that follow, it shall be seen that closed-form solutions for $\mathcal{I}_F$ cannot be obtained in all cases. Consequently, numerical integration is employed for approximation purposes. The specific routine employed here is called adaptive Gauss-Kronrod quadrature, with extrapolation by Wynn's ϵ-algorithm.

Finally, this chapter discusses the MCMC approach and the Metropolis-Hastings algorithm for simulation of the posterior distributions in Section 2.2.5.

Chapter 3

The Rayleigh distribution and its compounded forms

This chapter investigates the class of models chosen for the parametric survival analysis approach. The history and characteristics of the Rayleigh distribution are discussed, followed by the derivations of its modified forms. The Rayleigh is compounded with respect to both the exponential and Gamma distributions and furthermore, these new models are generalised. The chapter concludes with a discussion on the simulation study in which these models will be assessed, illuminating both the simulation procedure as well as the quantities of interest that will form the basis of the results.

3.1 The Rayleigh as a lifetime distribution

The Rayleigh distribution is a continuous probability model for a positive random variable. Denoting this variable by T, with an arbitrary realisation $t > 0$, its PDF has the form

$$f(t|\theta) = 2\theta t e^{-\theta t^2} \tag{3.1}$$

where $\theta > 0$ is the scale parameter. Accordingly, the CDF of the Rayleigh follows as

$$F(t) = P(T \leq t) = \int_0^t 2\theta x e^{-\theta x^2} dx = 1 - e^{-\theta t^2}. \tag{3.2}$$

Figure 3.1 visualises (3.1) for different values of the parameter. Other characteristics include the expected value

$$E[T] = \int_0^\infty t f(t|\theta) dt = \sqrt{\frac{\pi}{4\theta}}$$

which can be obtained using integrations by parts, as well as the variance

$$V[T] = E[T^2] - (E[T])^2 = \frac{4-\theta}{4\theta}.$$

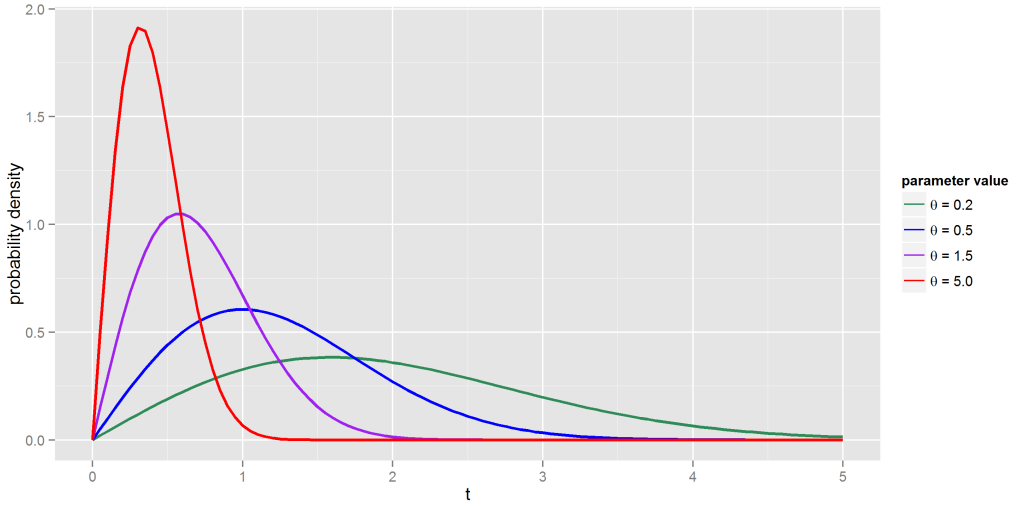


Figure 3.1: *Rayleigh density function for various values of the parameter θ.*

This distribution is named after John William Strut, or Lord Rayleigh, a prominent physicist and Nobel Laureate who made notable contributions to fields such as light scattering and acoustics. While doing work on random vibrations, Lord Rayleigh first pioneered the idea of the distribution that carries his name in the late 19th century (Lord Rayleigh, 1880), but it was only later that Siddiqui (1962) formalised the characteristics and properties of this distribution.

Interestingly enough, it can easily be shown that the magnitude R of a vector with coordinates Z_1 and Z_2, which are both iid standard normal variables, has a Rayleigh distribution, where $R = \sqrt{Z_1^2 + Z_2^2}$.

Nowadays, the Rayleigh distribution is useful for a very wide variety of purposes, from describing the background noise in magnetic resonance imaging (Sijbers et al., 1999), to modelling wave heights in oceanography (Casas-Prat and Holthuijsen, 2010). However, a significant application is in the context of survival analysis, since the Rayleigh

distribution is a popular choice to model time-to-event data. Bhattacharya and Tyagi (1990) note that the survival time of cancer patients has been observed to correspond to this distribution, as do Lalitha and Mishra (1996). Furthermore, many modifications of the Rayleigh distribution has been studied and applied to lifetime data, most notably Mostert et al. (1999) and Bekker et al. (2000). This will be discussed in Section 3.2.

One of the reasons why the Rayleigh distribution provides a good fit in certain survival analysis settings becomes clear when the important parameters are examined. From (3.2), the survival function for $t, \theta > 0$ is

$$S(t) = e^{-\theta t^2}$$

and from this, the hazard rate can be derived as

$$h(t) = 2\theta t. \tag{3.3}$$

It is immediately clear that the hazard rate is linearly increasing with time and consequently, a good choice to model lifetime data where ageing happens rapidly with time. Mostert et al. (1998) use the Rayleigh as a model for survival data with a Bayesian estimation procedure, as do others in more recent literature (Ahmed et al., 2013, Saleem and Aslam, 2009).

Another motivation for the attractiveness of the Rayleigh distribution as a survival model is its relation to the Weibull distribution. The form of the Weibull distribution makes it suitable to a broad range of applications, and it is especially common in parametric survival analysis (Klein and Moeschberger, 2003, Mudholkar et al., 1996). The form of its PDF, for $t > 0$ follows as

$$f(t|c, \theta) = c\theta t^{c-1} e^{-\theta t^c}, \tag{3.4}$$

with shape parameter $c > 0$ and scale parameter $\theta > 0$. Thus, the Rayleigh distribution (3.1) is a special case of the Weibull, with $c = 2$. The shape parameter of the Weibull makes it a suitable model for a larger class of survival data, since values of c smaller than, equal to and larger than 1 correspond to hazard rates that are respectively decreasing with time, constant with time and increasing with time.

In order to increase the scope of the Rayleigh distribution's application, some modifications to its form have been considered. Here, compounding and generalisation of the Rayleigh are invoked, with the goal of parameter enrichment and ultimately producing models with extended versatility and flexibility.

3.2 Compounding and generalising the Rayleigh distribution

3.2.1 Background of Rayleigh modifications

In Section 3.1, the Rayleigh distribution and its appeal as a model in parametric survival analysis were discussed. Modified versions derived from this distribution are now considered, forming the basis of this thesis.

The Rayleigh distribution (3.1) has a parameter controlling its scale and a corresponding hazard rate that increases linearly with time. Two types of modification that will expand the use of this distribution, compounding and generalisation, are investigated. Using these methods, the versatility of the original distribution is increased. The idea is that with additional model parameters, the modified Rayleigh distributions will lead to hazard rates which are more customisable to a given data set and applicable to a wider variety of lifetime data. Details of compounded and generalised models are discussed in Sections 3.2.1.1 and 3.2.1.2 respectively and in addition, derivations of the models that are considered here follow in Section 3.2.2.

The compound Rayleigh distribution with its unimodal hazard rate was introduced by Greenwich (1992), but studied in the framework of randomly censored survival data by Ghitany (2001), although without the use of Bayesian methods. Abdel-Ghaly and Attia (1993) introduce a model equivalent in nature to the compound Rayleigh with respect to the Gamma distribution, and investigates some of its characteristics. Approaches to broaden the scope of the Rayleigh model in similar ways can be found in the literature, specifically in Mostert et al. (1999) and Bekker et al. (2000), the foundations on which this thesis extends. These authors also investigate compounding with respect to the Gamma distribution and generalisation of Rayleigh models in the context of objective Bayesian analysis, as well as considering an array of loss functions for derivation of

Bayesian estimators. However, in order to simplify their modelling procedure, only one of the unknown model parameters that is to be estimated is considered continuous, the other is discretised, leading to priors and posteriors that have closed-form solutions in many cases. Here, their work is extended firstly by considering a wider array of compound Rayleigh models, since compounding with respect to the exponential distribution is also performed, and secondly by considering an additional asymmetric loss function with which estimators are derived. Most importantly, all unknown model parameters in the current work are assumed to be continuous, necessitating the use of numerical approximation methods, as described in Section 2.2.3.2, to evaluate some of the elements of the Fisher information matrices required for prior derivation.

Other ways in which the Rayleigh distribution can be extended that is loosely related to the work done here, is by considering two-component mixtures of the Rayleigh in a Bayesian context (Saleem and Aslam, 2008, Soliman, 2006), or by generalising the Rayleigh in different ways than the application in the current work (Cordeiro et al., 2013, Kundu and Raqab, 2005).

3.2.1.1 Compound distributions

Compounding increases the versatility of a PDF by transforming or replacing the parameters involved. This method shares some properties of the Bayesian methodology, since it amounts to assigning a probability distribution to the parameters in the parametric model. Flexibility is gained through compounding by allowing a singe-parameter probability model to have multiple parameters controlling its form. The additional parameter(s) is responsible for the heavier tails of the compound distribution. Some work on the definition and characteristics of these distributions have been done by Gurland (1957) and McDonald and Butler (1987).

The following general notation will be used. Suppose that a random variable T is distributed according to a probability distribution F with PDF f, dependent on unknown parameter(s) θ. Furthermore, suppose that θ is itself distributed to some distribution G with PDF g, dependent on hyper-parameter(s) γ. Compounding now takes place when F is marginalised over G. The resulting distribution for T is referred to as the compound distribution of F with respect to G and is now independent of the original model parameter(s) θ. If the original distribution of T is denoted by $f(t|\theta)$, and θ is regarded

as a random variable with distribution $g(\theta|\gamma)$, the compound distribution follows as

$$f(t|\gamma) = \int_{\Theta} f(t|\theta)g(\theta|\gamma)d\theta,$$

where Θ is the range of the original parameter θ

Compound probability distributions can also be formulated from the point of view of mixture models (Grubbstrom and Tang, 2006). These are models for which the probability distribution of the variable consists of a weighted sum of distributions, called the mixture components. For example, we can construct a mixture PDF for a random variable T, with component PDFs $f_1(t)$ and $f_2(t)$ and mixture weights w_1 and w_2 as follows:

$$f_{\text{mix}}(t) = \sum_{i=1}^{2} w_i f_i(t).$$

Compound models now emerge as mixture models with an infinite number of component distributions, $f(t|\theta)$, where all values of θ in some set Θ are considered. The mixture weights are determined by a new probability distribution with its own parameter, $g(\theta|\gamma)$. Finally, for mathematical tractability, the infinite sum is replaced by an integral over all values Θ. The resulting distribution becomes

$$\int_{\Theta} f(t|\theta)g(\theta|\gamma)d\theta = f(t|\gamma),$$

the compound model.

3.2.1.2 Generalised distributions

Generalisation of probability distributions as an approach to broaden the versatility of a model is not as well defined as compounding. In general, it refers to the inclusion of a new parameter into the model, such that a specific value of this parameter will yield the original distribution. The additional variable increases the flexibility of the generalised distributions to take on a wider range of forms. It should be noted, however, that there is no single "correct" way in which this method can be applied. Three distinct generalisations of the Rayleigh distribution will be discussed here – two examples which can be found in literature, as well as the form that will be used in this thesis, described in Section 3.2.2.3. Note that in the context of parametric survival analysis, the

generalisation can be brought about at either the level of the probability distributions or the hazard rate, since the former can be derived from the latter. In the following discussions, the models are generalised and from there, relevant characteristic functions are inferred.

Kundu and Raqab (2005) set out different methods of parameter estimation for a distribution that was originally named the *Burr Type X* distribution, but termed a generalised Rayleigh by Surles and Padgett (2001). The CDF and PDF of that model respectively have the following forms, for $t, \alpha, \lambda > 0$:

$$F(t|\alpha,\lambda) = \left(1 - e^{-\lambda t^2}\right)^{\alpha},$$
$$f(t|\alpha,\lambda) = 2\alpha\lambda t e^{-\lambda t^2}\left(1 - e^{-\lambda t^2}\right)^{\alpha-1}.$$

The same authors more recently generalised this distribution further to include a location parameter μ, such that for $t > \mu$,

$$F(t|\alpha,\lambda,\mu) = \left(1 - e^{-\lambda(t-\mu)^2}\right)^{\alpha},$$

although this work is based around the estimation of the so-called stress-strength parameter $R = P(Y < X)$, where Y and X are distributed according to their three-parameter generalised Rayleigh model (Kundu and Raqab, 2015). Additional work was also done on their two-parameter model in recent years, expanding the scope of estimation techniques and showing an application to failure times of machines (Al-Kanani and Jasim, 2014).

A second case of a generalised Rayleigh distribution in the literature is used by Cordeiro et al. (2013). The forms of the CDF and PDF of the model they work with can be written respectively as

$$F(t|\theta,\alpha) = \frac{1}{\Gamma(\alpha)}\int_0^{\theta t^2} x^{\alpha-1}e^{-x}dx,$$
$$f(t|\theta,\alpha) = \frac{2\theta^{\alpha}}{\Gamma(\alpha)}t^{2\alpha-1}e^{-\theta t^2},$$

for $t, \theta, \alpha > 0$.

Even though the Gamma function in this model's specification already makes it quite difficult to work with mathematically, they further expand it with two more parameters

to what they call the Beta generalised Rayleigh distribution. For brevity, this new form is not given here, but despite the complexity, properties of this distribution are derived and an application to survival data in the field of engineering showcases its intended use. Additionally, other work has also been done to expand on the form given above, such as the derivation of the Kumaraswamy generalised Rayleigh distribution, which can be regarded as a mixture distribution of generalised Rayleigh densities (Gomes et al., 2014).

3.2.2 Compounding and generalising the Rayleigh model

The approaches considered in this thesis are largely based on the studies of Mostert et al. (1999) and Bekker et al. (2000). In their work, the Rayleigh is compounded with respect to the Gamma distribution and furthermore, flexibility is increased through a generalisation. Instead of squaring survival times in the form of the PDF, a general power c is considered. A similar approach is followed in the current work, with the addition of investigating the exponential distribution in the compounding process. This is a special case of the Gamma distribution for which the shape parameter is unity. These distributions are mainly chosen for the mathematically attractive form of the resulting compounded models. Apart from increasing flexibility, one can also conceptualise compounding of the Rayleigh model by envisioning a scenario where subjects are drawn from a population with variable hazard rates. Assigning a probability distribution to the parameter(s) through compounding leads to the notion of a random hazard rate, and the subsequent survival times will then be distributed according to a compound Rayleigh model.

The hazard rates for the models of interest are derived in the Chapters 4 and 5. In equations (4.2), (4.8), (5.1) and (5.8), it can be seen that the hazard rate is no longer monotonically increasing with time, but has a stationary point corresponding to a peak time of maximum hazard. This class of hazard rates can be useful when this peak time is an important outcome (Greenwich, 1992). The stationary points inevitably lead to a decreasing hazard rate in later time periods, but this type of hazard curve can still be of value when only the initial time periods are of interest. Moreover, the use of a decreasing hazard rate can be justified with a conditional argument (Abdel-Ghaly and Attia, 1993).

3.2.2.1 Compounding with respect to the exponential distribution

First, the Rayleigh PDF (3.1) is compounded with the exponential distribution, such that

$$g(\theta|\gamma) = \gamma e^{-\gamma\theta}, \qquad \gamma > 0.$$

The new model shall be referred to as the compound Rayleigh with respect to exponential (CRE) model and be formulated in terms of the parameter γ. Its PDF is obtained by using the procedure in Section 3.2.1.1, leading to

$$f(t|\gamma) = 2\gamma t \int_0^\infty \theta e^{-\theta(t^2+\gamma)} d\theta.$$

Using integration by parts, one obtains the simplified form

$$f(t|\gamma) = 2\gamma t(t^2+\gamma)^{-2}. \tag{3.5}$$

Figure 3.2 illustrates how the shape of the new distribution changes by varying the γ parameter.

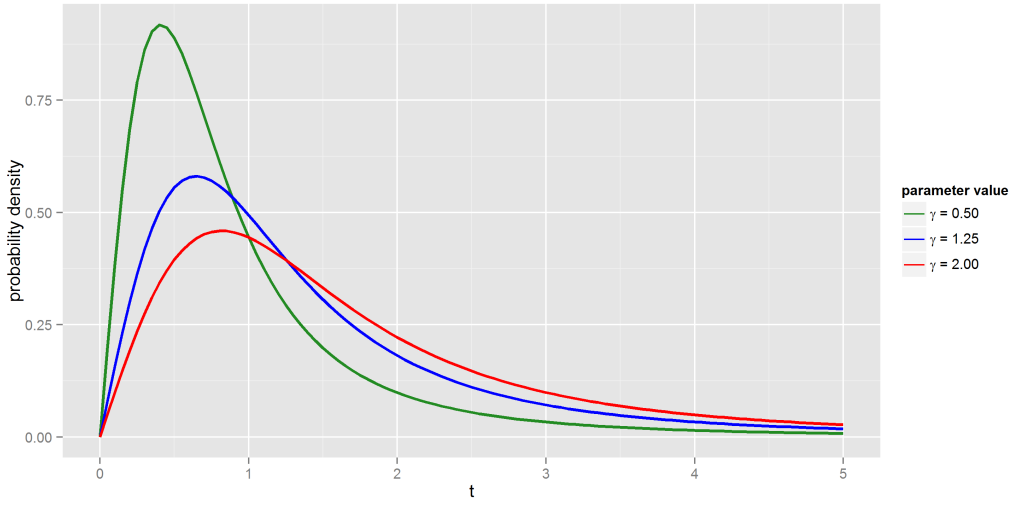


Figure 3.2: *CRE density function for various values of parameter γ.*

The CRE distribution function can be found through integration of (3.5), for $t > 0$, it follows that

$$\begin{aligned} F(t|\gamma) &= \int_0^t 2\gamma x(x^2+\gamma)^{-2}dx \\ &= 1 - \frac{\gamma}{t^2+\gamma} \\ &= 1 - \left(1 + \frac{t^2}{\gamma}\right)^{-1}. \end{aligned} \tag{3.6}$$

3.2.2.2 Compounding with respect to the Gamma distribution

Secondly, the Rayleigh PDF (3.1) is compounded with respect to the Gamma distribution, resulting in what will be referred to as the compound Rayleigh with respect to Gamma (CRG) model, with two model parameters.

The parameter θ is now assumed to have a Gamma distribution with parameters $\alpha > 0$, the shape, and $\beta > 0$, the scale, such that

$$g(\theta|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\theta^{\alpha-1}e^{-\beta\theta}.$$

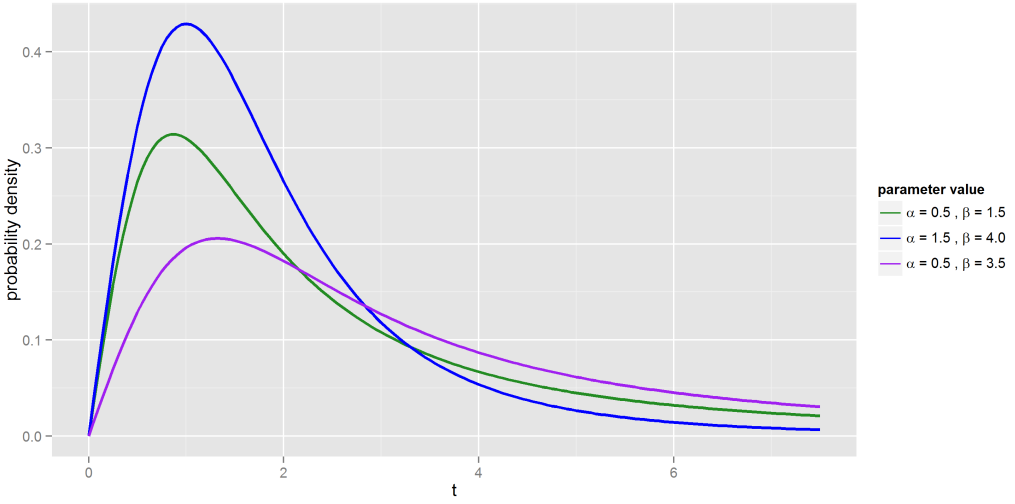


Figure 3.3: *CRG density function for various values of parameters α and β.*

The compound PDF is obtained by integrating out θ and one finds

$$\begin{aligned} f(t|\alpha,\beta) &= \int_0^\infty \frac{2\beta^\alpha}{\Gamma(\alpha)} t\theta^\alpha e^{-\theta(t^2+\beta)} d\theta \\ &= \frac{2t\beta^\alpha}{\Gamma(\alpha)} \frac{\Gamma(\alpha+1)}{(t^2+\beta)^{\alpha+1}} \\ &= 2t\alpha\beta^\alpha (t^2+\beta)^{-(\alpha+1)} \end{aligned} \tag{3.7}$$

with simplification of the Gamma function. Figure 3.3 shows how the form of this PDF changes as the shape and scale varies.

Integration of (3.7) yields the distribution function of the CRG model,

$$F(t|\alpha,\beta) = 1 - \left(1 + \frac{t^2}{\beta}\right)^{-\alpha}. \tag{3.8}$$

3.2.2.3 Generalising the compounded distributions

In Section 3.2.1.2, some approaches to generalise the Rayleigh distribution was discussed. The approach relevant to the current work, based on Bekker et al. (2000), is now considered.

Accordingly, the compounded models derived in the previous sections are enriched with an additional shape parameter c. Considering the CDFs of the CRE and CRG distributions in equations (3.6) and (3.8) respectively, the generalisation is applied specifically to the exponent of the survival time t. Consequently, the CDF of the generalised CRE model becomes

$$F(t|\gamma,c) = 1 - \left(1 + \frac{t^c}{\gamma}\right)^{-1}, \tag{3.9}$$

and the CDF of the generalised CRG model becomes

$$F(t|\alpha,\beta,c) = 1 - \left(1 + \frac{t^c}{\beta}\right)^{-\alpha}. \tag{3.10}$$

These distributions shall henceforth be referred to as the GCRE and GCRG models. Their corresponding PDFs can now easily be obtained with differentiation. For the GCRE model, this amounts to

$$f(t|\gamma,c) = c\gamma t^{c-1}(t^c+\gamma)^{-2}, \tag{3.11}$$

with the form for various parameter configurations portrayed in Figure 3.4.

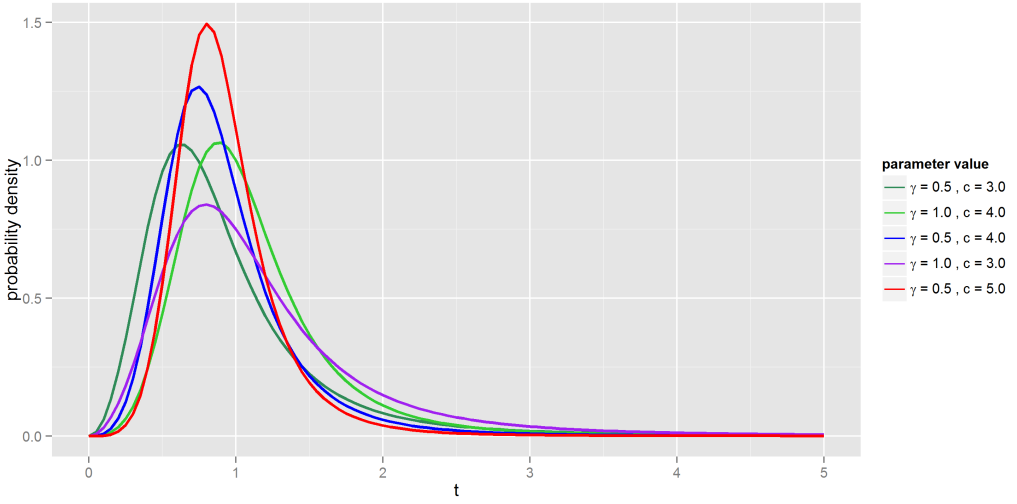


Figure 3.4: *GCRE density function for various values of parameters γ and c.*

Similarly, the PDF of the GCRG model is found to be

$$f(t|\alpha,\beta,c) = \alpha c \beta^{\alpha} t^{c-1}(t^c+\beta)^{-(\alpha+1)} \tag{3.12}$$

and Figure 3.5 below shows examples of the form for some parameter configurations.

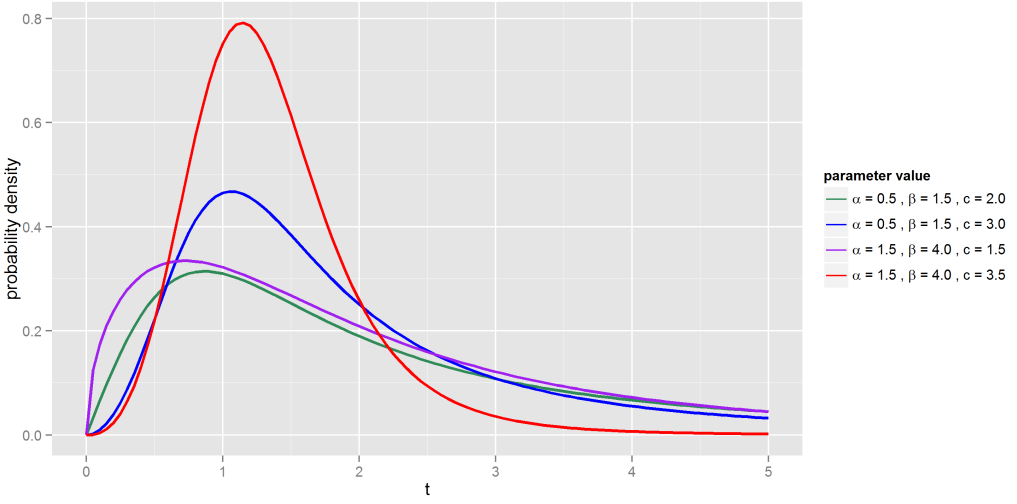


Figure 3.5: *GCRG density function for various values of parameters α, β and c.*

3.2.3 Expansion and estimation of the compounded Rayleigh models

Section 3.2.2.1 presents the derivations of the four compound Rayleigh models investigated in this thesis. This includes the Rayleigh models compounded with respect to the exponential and Gamma distributions, respectively (3.5) and (3.7), and their generalised counterparts, respectively (3.11) and (3.12).

In the Chapters 4 and 5, the properties of these models that are of interest in a parametric survival setting will be considered in more detail. In particular, the survival functions, hazard rates and likelihood functions will be elicited from the models' forms. Thereafter, the Bayesian approach to parameter estimation will be followed, described in Section 2.2, and the Fisher information, non-informative priors and Bayesian estimators will be derived for each model. Finally, the four models will be subjected to simulation studies in which their performance will be assessed, the topic pursued in the remainder of this chapter.

3.3 Simulation studies on the compounded Rayleigh models

The four Rayleigh models of interest, derived in the previous section, are applied and investigated in simulation studies. In Section 2.2.5, MCMC and specifically the Metropolis-Hastings algorithm was considered as a way to generate realisations from a posterior distribution when a closed-form solution is not available. Chapters 4 and 5 will reveal that the likelihoods and non-informative priors for the compound Rayleigh models lead to posterior distributions which are only specified in proportional form, prompting the use of the aforementioned methods.

The simulation studies consists of two aspects. First, Section 3.3.1 discusses the generation of random samples from the compound Rayleigh models. Thereafter, Section 3.3.2 describes the procedure followed to calculate and assess the performance of each model, based on realisations from the posterior distributions. Both of these aspects, at their cores, depend on the generation of random $U(0,1)$ variates. In all cases where applicable, the powerful Mersenne Twister algorithm is relied on for this purpose (Matsumoto and

Nishimura, 1998). All parts of the simulation study were coded with the R statistical computing platform, including the visualisations of the results (R Core Team, 2012).

3.3.1 Simulating random samples from Rayleigh models

In order to apply MCMC techniques to simulate realisations from the posterior distributions, random samples from the compound Rayleigh models need to be simulated. In this way, exact characteristics of the probability mechanism that the data is generated from is known. Although this knowledge is obviously unavailable in practice, it allows for an accurate comparison of the results of different Bayesian estimators in this study.

The CDFs of the four Rayleigh models are explicitly known, therefore the inverse transform method can be used for generation of random samples. This is a standard approach and is briefly described in Appendix A.1. The resulting transformations with which samples from the models are generated, are summarised in Table 3.1. Note that u is a randomly drawn uniformly distributed $U(0,1)$ variate.

Table 3.1: *Transformations used with inverse transform method to draw samples from compound Rayleigh models.*

model	CDF	transformation
CRE	(3.6)	$t = \sqrt{\gamma(u^{-1}-1)}$
CRG	(3.8)	$t = \sqrt{\beta(u^{-\frac{1}{a}}-1)}$
GCRE	(3.9)	$t = \left\{\gamma(u^{-1}-1)\right\}^{\frac{1}{c}}$
GCRG	(3.10)	$t = \left\{\beta(u^{-\frac{1}{a}}-1)\right\}^{\frac{1}{c}}$

The generated samples are used as analogues to lifetime data. Thus, a level of censoring is specified during the simulation study. The proportion of non-censored values is denoted by δ. Censoring is now simulated by drawing an amount of random $U(0,1)$ variates, $u_1, u_2, \ldots, u_n$, where n is the sample size. Each observation i in the sample is then marked as censored if $u_i > \delta$, or marked as an event otherwise.

3.3.2 Simulation study methods and quantities of interest

This section describes the simulation study that each of the four compound Rayleigh models, derived in Section 3.2.2, undergoes. In very brief terms, a large number of data samples are generated from the model with known parameters. Each sample is then used

with MCMC to obtain realisations from the posterior distribution of the parameters and depending on the model, this is carried out for an array of non-informative prior distributions. Lastly, performance is assessed by calculating common summary statistics such as the posterior mean and variance, credible intervals, uncertainty measures such as the mean squared error (MSE) and bias. Four variants of Bayesian estimators, derived assuming different loss functions, are also evaluated.

In the simulation study, the measures used to assess the performance are the MSE and mean absolute error (MAE) for quantification of the precision, as well as the bias for quantification of the accuracy. Note that the term "precision" is used here to refer to the inverse of variance, thus a high degree of precision is desirable. The MSE can be defined for an arbitrary estimator $\hat{\theta}$ of the true value θ, to be

$$\begin{aligned} \text{MSE}(\hat{\theta}) &= \text{E}_\theta\left[(\hat{\theta}-\theta)^2\right] \\ &= \text{V}_\theta[\hat{\theta}] + \left\{\text{bias}(\hat{\theta})\right\}^2. \end{aligned}$$

Thus, it takes both the variance and bias into account. The MAE can be defined in the same way, but by using the absolute difference as an alternative to the squared difference. The absolute bias can be calculated from the MSE with the formula above, but can also be defined independently as

$$\text{bias}(\hat{\theta}) = \text{E}_\theta\left[\hat{\theta}-\theta\right].$$

This definition of the bias is used in order to obtain its direction in addition to its magnitude. The calculation of the MSE, MAE and bias are explained below. A last performance measure is the *coverage*. This is defined as the proportion of 95% credible intervals that contain the true parameter value. The coverage is inherently a frequentist measure, since with Bayesian statistics, the only concern is conditioning on a single sample. Notwithstanding, it is still interesting to observe how the coverage deviates from the expected 95% level for different simulation procedures.

A more detailed stepwise description of the simulation procedure now follows.

1. For the compound Rayleigh model under consideration, a sample of size n is generated with a level of censoring δ and known parameter value(s) denoted by θ, as described in Section 3.3.1.

2. The sample is used in the Metropolis-Hastings algorithm to draw 12500 realisations of the parameter(s) from the posterior distribution(s). The first 2500 are discarded in accordance with Section 2.2.5.2 and the remaining 10000 regarded in the consequent calculations. For the CRE model and its generalised counterpart, only one posterior is considered, corresponding to the Jeffreys prior. For the CRG model and its generalised counterpart, three distinct posteriors are available, corresponding to the three non-informative priors discussed in Section 2.2.4. In the interest of comparison, 10000 realisations are drawn from each of these posteriors.

3. The realisations from the posterior distributions of the parameters are used to compute Bayesian estimators under four different loss functions, namely AE loss, SE loss, LINEX loss and GE loss, as discussed in Section 2.2.2. These include the posterior mean and median. Additionally, the 2.5^{th} and 97.5^{th} percentiles for credible interval bounds, as well as the posterior variance, are calculated and stored. The law of large numbers permits the expected values to be estimated by arithmetic means across the 10000 realisations.

4. Steps 1 – 3 are now repeated 1000 times, such that a large number of distinct random samples are considered, each with the same n, δ and true parameter values. Summary measures can now be acquired by averaging all Bayesian estimates. In addition, measures of accuracy and precision can also be calculated. For each estimator, the MSE, MAE and bias now become

$$\begin{aligned}
\text{MSE}(\hat{\theta}) &= \frac{1}{1000}\sum_{i=1}^{1000}\left(\hat{\theta}_i - \theta\right)^2 \\
\text{MAE}(\hat{\theta}) &= \frac{1}{1000}\sum_{i=1}^{1000}\left|\hat{\theta}_i - \theta\right| \\
\text{bias}(\hat{\theta}) &= \frac{1}{1000}\sum_{i=1}^{1000}\left(\hat{\theta}_i - \theta\right).
\end{aligned}$$

   The coverage can also be calculated as the proportion of 95% credible intervals that contain the true parameter value. These results can now be presented visually and interpreted.

5. All previous steps are repeated with a different δ value. In this way, complete results are gathered for two levels of censoring and the effect of censoring can

clearly be seen. Throughout the simulation studies, $\delta = 1$ (no censoring) and $\delta = 0.8$ (20% censoring) were considered.

6. Lastly, for a given compound Rayleigh model, all previous steps are repeated with different parameter configurations. This allows the interpretations and conclusions drawn from results to be as general as possible.

In the chapters that follow, the simulation procedure outlined above will be carried out numerous times. Note that in all cases, a sample size of $n = 50$ was used. However, to investigate the effect of varying sample size, one parameter configuration was chosen for each model with which a sample size of $n = 30$ was also considered.

3.4 Summary

An overview of the class of models of interest, investigated in detail in this chapter, now follows.

The Rayleigh distribution, with PDF in equation (3.1),

$$f(t|\theta) = 2\theta t e^{-\theta t^2},$$

is a popular choice for parametric survival analysis, especially when the hazard rate increases sharply with time.

The Rayleigh is compounded with respect to the exponential distribution, leading to the so-called CRE model, as well as with respect to the Gamma distribution, leading to the CRG model. Furthermore, these models are made more flexible through a generalisation, leading to the GCRE and GCRG models respectively. The forms of the density functions of these models are summarised below, using equations (3.5), (3.7), (3.11) and (3.12).

model name	PDF
CRE	$f(t\|\gamma) = 2\gamma t(t^2+\gamma)^{-2}$
CRG	$f(t\|\alpha,\beta) = 2t\alpha\beta^{\alpha}(t^2+\beta)^{-(\alpha+1)}$
GCRE	$f(t\|\gamma,c) = c\gamma t^{c-1}(t^c+\gamma)^{-2}$
GCRG	$f(t\|\alpha,\beta,c) = \alpha c\beta^{\alpha}t^{c-1}(t^c+\beta)^{-(\alpha+1)}$

In Section 3.3.2, the simulation study methods and quantities of interest are discussed in some detail. For each of the four models under consideration, the steps of the simulation study can be contracted as follows:

1. Simulate sample of n survival times, with level of censoring δ and parameter configuration θ.
2. Conditional upon this sample, generate 10000 realisations from the three posteriors corresponding to the non-informative priors considered.
3. Calculate distributional properties of posteriors (Bayesian estimators under all loss functions, as well as credible intervals and posterior variance).
4. Repeat steps 1–3 1000 times, keeping n, δ and θ constant, enabling calculation of measures of accuracy and precision (MSE, MAE, bias) and frequentist properties (coverage).
5. Repeat steps 1–4 with a different value of δ.
6. Repeat steps 1–5 for different parameter configurations θ.

Two levels of censoring are investigated, i.e. $\delta = \{0, 0.2\}$, as well as two sample sizes, $n = 50$ and $n = 30$, although the latter is only investigated partially and not for all parameter configurations.

Chapter 4

Simulation study of compound models

This chapter portrays the simulation study using the Rayleigh models compounded with respect to the exponential and Gamma distributions, i.e. the CRE and CRG models. Their survival and hazard functions, likelihood functions, Fisher information matrices and Bayesian estimators are derived, as well as the relevant non-informative prior distributions. Thereafter, the simulation study results are shown and discussed.

4.1 The CRE model

4.1.1 Model characteristics

The CRE distribution and density functions were derived for a non-negative lifetime variable $t > 0$ in Section 3.2.2.1, resulting in a model with a single parameter γ. For survival analysis, the two main functions of interest are the survival function (2.1) and the hazard rate (2.2), even though estimation of only the latter will be exemplified here. They can easily be obtained from the CDF (3.6), such that

$$S(t,\gamma) = \left(1 + \frac{t^2}{\gamma}\right)^{-1} \tag{4.1}$$

and

$$\text{and } h(t,\gamma) = \frac{2t}{t^2+\gamma}. \tag{4.2}$$

The form of the hazard rate is presented graphically in Figure 4.1. It is not a monotonic function of time, contrary to that of the Rayleigh model in (3.3).

The Fisher information is required for the derivation of some of the prior distributions. Considering the natural log of (3.5), denoted here by l_f, the second partial derivative with respect to γ becomes

$$\begin{aligned} l_f &= \ln f(t|\gamma) = \ln 2 + \ln \gamma + \ln t - 2\ln(t^2+\gamma) \\ \Rightarrow \frac{\partial^2 l_f}{\partial \gamma^2} &= -\frac{1}{\gamma^2} + \frac{2}{(t^2+\gamma)^2}. \end{aligned}$$

Since $\mathcal{I}_F$ is the negative of the expected value of $\frac{\partial^2 l_f}{\partial \gamma^2}$, it is necessary to calculate the expectation with respect to the compound PDF (3.5). This follows as

$$\mathrm{E}_T\left[\frac{1}{(T^2+\gamma)^2}\right] = \int_0^\infty 2t\gamma(t^2+\gamma)^{-4}dt = \frac{1}{3\gamma^2}.$$

and from (2.11), the Fisher information therefore becomes

$$\mathcal{I}_F = \mathrm{E}_T\left[\frac{1}{\gamma^2} - \frac{2}{(T^2+\gamma)^2}\right] = \frac{1}{\gamma^2} - \frac{2}{3\gamma^2} = \frac{1}{3\gamma^2}. \tag{4.3}$$

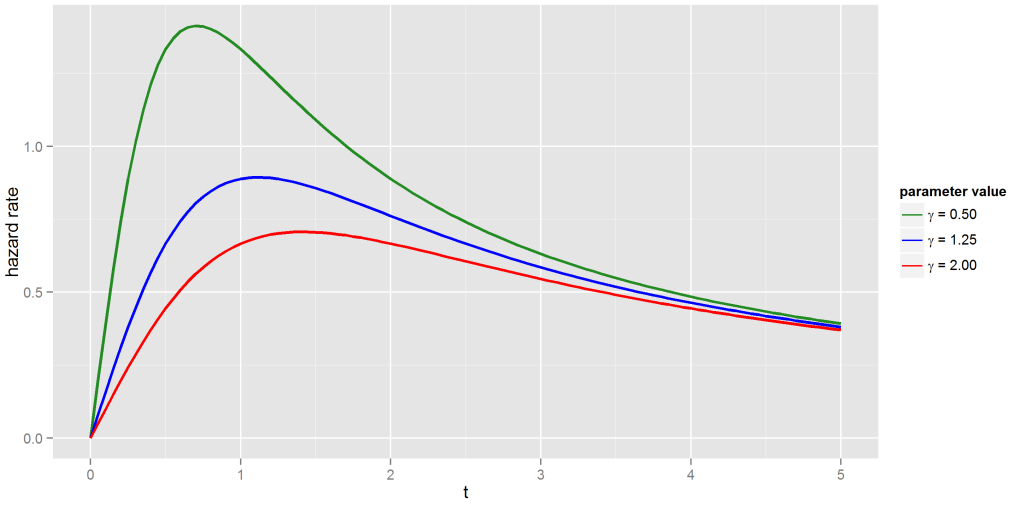


Figure 4.1: *Hazard rate of CRE model for various values of its parameter.*

4.1.2 Prior and posterior distributions

In order to obtain the posterior distribution with which analysis regarding the parameter can be performed, the likelihood and a prior distribution for γ are required. The CRE model has a single parameter, therefore the only concern is the derivation of the Jeffreys prior, since the reference and PM priors are equivalent in the univariate case.

The Jeffreys prior, π_{jeff}, discussed in Section 2.2.4.1, is defined to be proportional to the square root of $\mathcal{I}_F$. Here, equation (4.3) is used to obtain the proportional form of the Jeffreys prior,

$$\begin{aligned}\pi_{\text{jeff}}(\gamma) &\propto \sqrt{\frac{1}{3\gamma^2}} \\ &\propto \frac{1}{\gamma}.\end{aligned} \tag{4.4}$$

Considering now a sample of n survival time observations $\mathbf{t} = (t_1, t_2, \ldots, t_n)$, ordered such that the first d are non-censored and the remaining $(n-d)$ right censored, the likelihood function follows from (2.3) and becomes

$$\begin{aligned}\mathcal{L}(\gamma|\mathbf{t}) &\propto \prod_{i=1}^{d} f(t_i|\gamma) \prod_{j=d+1}^{n} S(t_j) \\ &\propto \prod_{i=1}^{d} \frac{\gamma t_i}{(t_i^2+\gamma)^2} \prod_{j=d+1}^{n} \frac{\gamma}{t_j^2+\gamma} \\ &\propto \gamma^n \prod_{i=1}^{d} \frac{t_i}{t_i^2+\gamma} \prod_{j=1}^{n} \frac{1}{(t_j^2+\gamma)}.\end{aligned}$$

From a computational point of view, it is better to reformulate the likelihood in terms of natural logarithms. Consequently, define

$$W_1(\gamma) = \ln\left(\prod_{i=1}^{d} \frac{t_i}{t_i^2+\gamma}\right) = \sum_{i=1}^{d} \ln\left(\frac{t_i}{t_i^2+\gamma}\right) \tag{4.5}$$

$$W_2(\gamma) = -\ln\left(\prod_{j=1}^{n} \frac{1}{t_j^2+\gamma}\right) = \sum_{j=1}^{n} \ln(t_j^2+\gamma) \tag{4.6}$$

and then the proportional form of the likelihood function can be written as

$$\mathcal{L}(\gamma|\mathbf{t}) \propto \gamma^n e^{W_1(\gamma)-W_2(\gamma)}.$$

Hence, using (4.4), the proportional form of the posterior distribution for γ becomes

$$\begin{aligned}\pi_{\text{jeff}}(\gamma|\mathbf{t}) &\propto \pi_{\text{jeff}}(\gamma) \cdot \mathcal{L}(\gamma|\mathbf{t}) \\ &\propto \gamma^{n-1} e^{W_1(\gamma)-W_2(\gamma)},\end{aligned} \tag{4.7}$$

with definitions (4.5) and (4.6) above. This form will be used in the MCMC simulation to generate realisations of γ from its posterior distribution once Bayesian estimators for γ are defined.

4.1.3 Bayesian estimators of the parameter

Following the acquisition of the posterior distribution (4.7) for the parameter γ, its Bayesian estimators can be formally stated. General Bayesian estimators for the loss functions under consideration were derived in Section 2.2.2.3. Considering the SE loss function, the estimator for γ is the posterior expected value

$$\hat{\gamma}_{\text{SE}} = \text{E}_{\gamma|\mathbf{t}}[\gamma].$$

Under absolute error (AE) loss, the estimator becomes the posterior median

$$\hat{\gamma}_{\text{AE}} = \text{median}_{\gamma|\mathbf{t}}[\gamma].$$

Using the asymmetric LINEX loss function with its hyper-parameter a, the Bayesian estimator is given by

$$\hat{\gamma}_{\text{LNX(a)}} = -\frac{1}{a} \ln \text{E}_{\gamma|\mathbf{t}} \left[e^{-a\gamma}\right],$$

and lastly, using the asymmetric GE loss function with its hyper-parameter k, the Bayesian estimator is

$$\hat{\gamma}_{\text{GE(k)}} = \left(\text{E}_{\gamma|\mathbf{t}} \left[\gamma^{-k}\right]\right)^{-\frac{1}{k}}.$$

The derivation of Bayesian estimators for functions of the parameter of interest yield analogous results to those above. Thus, the estimators under different loss functions for

the survival function become

$$\begin{aligned}
\hat{S}_{\text{SE}}(t,\gamma) &= \text{E}_{\gamma|\mathbf{t}}[S(t,\gamma)] \\
\hat{S}_{\text{AE}}(t,\gamma) &= \text{median}_{\gamma|\mathbf{t}}[S(t,\gamma)] \\
\hat{S}_{\text{LNX(a)}}(t,\gamma) &= -\frac{1}{a}\ln \text{E}_{\gamma|\mathbf{t}}\left[e^{-aS(t,\gamma)}\right] \\
\hat{S}_{\text{GE(k)}}(t,\gamma) &= \left(\text{E}_{\gamma|\mathbf{t}}\left[S(t,\gamma)^{-k}\right]\right)^{-\frac{1}{k}},
\end{aligned}$$

from (4.1) and similarly, those for the hazard rate become

$$\begin{aligned}
\hat{h}_{\text{SE}}(t,\gamma) &= \text{E}_{\gamma|\mathbf{t}}[h(t,\gamma)] \\
\hat{h}_{\text{AE}}(t,\gamma) &= \text{median}_{\gamma|\mathbf{t}}[h(t,\gamma)] \\
\hat{h}_{\text{LNX(a)}}(t,\gamma) &= -\frac{1}{a}\ln \text{E}_{\gamma|\mathbf{t}}\left[e^{-ah(t,\gamma)}\right] \\
\hat{h}_{\text{GE(k)}}(t,\gamma) &= \left(\text{E}_{\gamma|\mathbf{t}}\left[h(t,\gamma)^{-k}\right]\right)^{-\frac{1}{k}},
\end{aligned}$$

from (4.2).

4.1.4 Simulation results

A simulation study was performed to assess the accuracy and precision of the Bayesian estimators for the parameter γ and the hazard function of the CRE model. The methodology is described in detail in Section 3.3.2. The simulation procedure was carried out for three different values of the parameter: $\gamma = \{0.5, 1.25, 2\}$, and two levels of censoring: $\delta = \{1, 0.8\}$. A sample size of 50 was used for the simulated survival time observations, except for the parameter choice $\gamma = 1.25$, where sample sizes of both 30 and 50 were considered.

A graphical presentation of the results allows for convenient comparison of the different simulation configurations. To this end, four different types of plots are shown for each parameter and censoring configuration:

- The first type of plot showcases the performance of the parameter's Bayesian point estimates. The x-axis demarcates the value of γ, with vertical lines of different colours drawn for the estimates $\hat{\gamma}_{\text{AE}}$ (green) and $\hat{\gamma}_{\text{SE}}$ (blue). In addition, orange

and red line graphs show how the values of respectively $\hat{\gamma}_{\text{LNX(a)}}$ and $\hat{\gamma}_{\text{GE(k)}}$ change as functions of their hyper-parameters, both portrayed on the y-axis.

- The second type of plot exhibits the MSE versus the bias. Here, the MSE and bias were calculated for various values of the parameter, with the true value clearly marked in red.

- The coverages for different parameter values are also displayed, with a dashed black line indicating the expected 95% level. This summarises the frequentist properties of the simulation.

- Bayesian estimators for functions of the model parameter are also of interest. For brevity, only estimates of the hazard rate are plotted against the true function. To increase the clarity of these plots, estimators corresponding to symmetric and asymmetric loss functions are separated. The latter is illustrated by choosing only one value for the symmetry parameters in each case. These values were chosen to roughly agree with those where $\hat{\gamma}_{\text{LNX(a)}}$ and $\hat{\gamma}_{\text{GE(k)}}$ were closest to reality. Note that while the Bayesian estimates of γ where averaged over the 1000 samples, only one sample is considered to exemplify the estimation of the hazard rate over time.

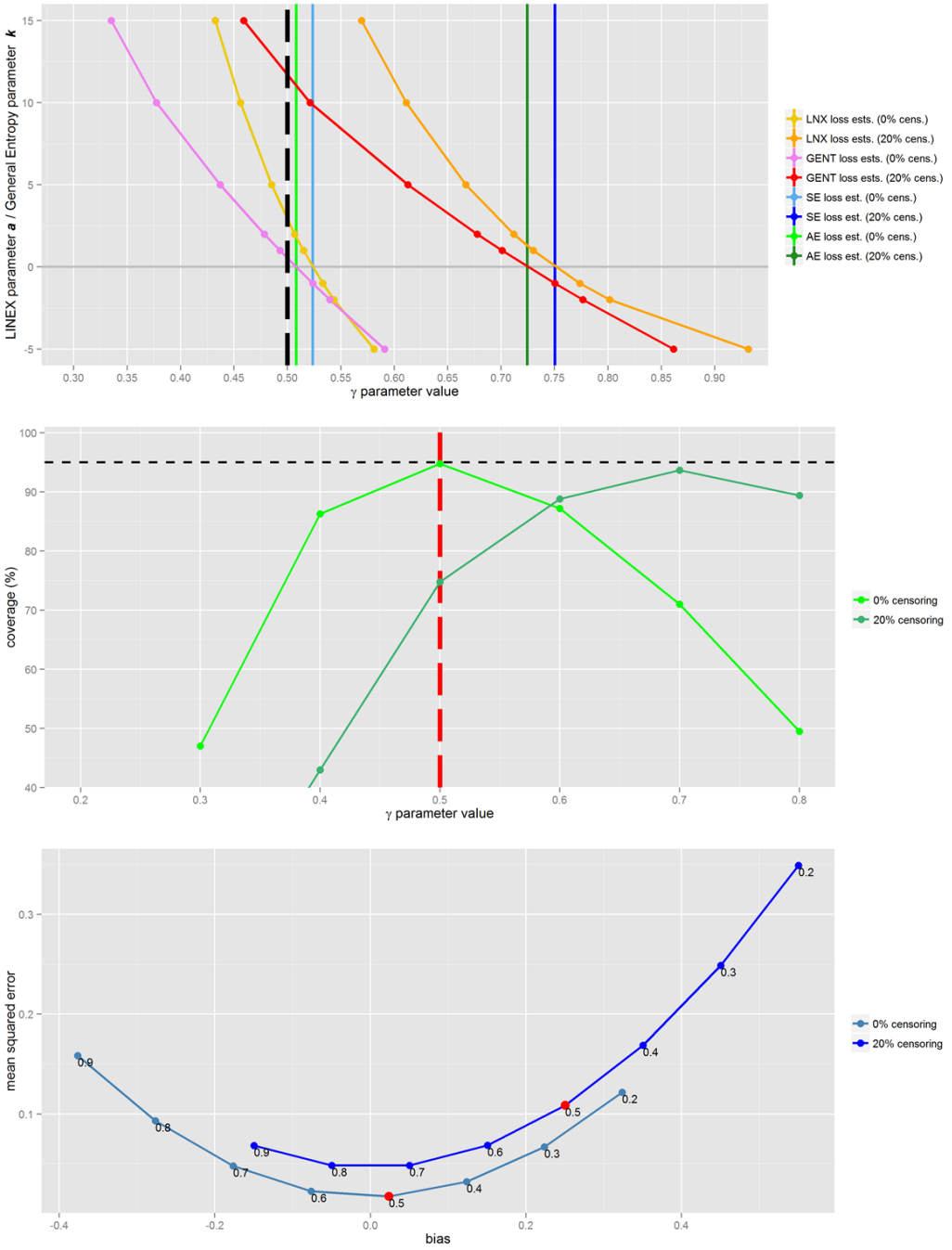


Figure 4.2: *Plots of Bayesian estimates (top), coverage (middle) and MSE against bias (bottom) for CRE model with* $\gamma = 0.5$*, and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours).*

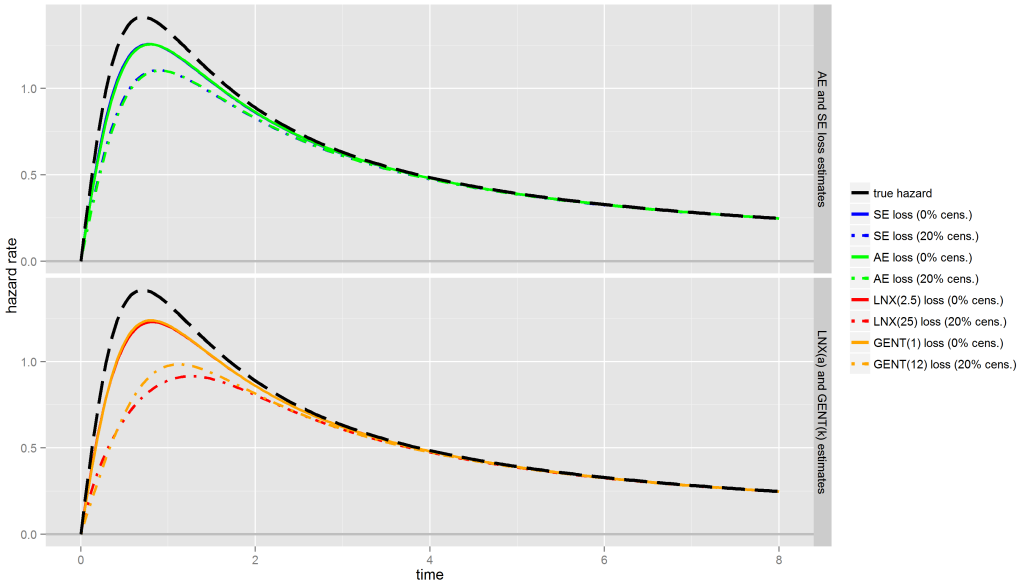


Figure 4.3: *Plots of Bayesian estimates of the hazard function for the CRE model with* $\gamma = 0.5$ *and two levels of censoring, derived using four different loss functions.*

Table 4.1: *The MAE, MSE and bias of estimators for the CRE model, with parameter* $\gamma = 0.5$.

estimator	MAE	MSE	bias
0% censoring			
$\hat{\gamma}_{\text{AE}}$	0.0980	0.0160	0.0082
$\hat{\gamma}_{\text{SE}}$	0.1010	0.0174	0.0238
$\hat{\gamma}_{\text{LNX(-2)}}$	0.1105	0.0218	0.0429
$\hat{\gamma}_{\text{LNX(2)}}$	0.0953	0.0150	0.0064
$\hat{\gamma}_{\text{GE(-2)}}$	0.1058	0.0197	0.0393
$\hat{\gamma}_{\text{GE(2)}}$	0.0969	0.0148	-0.0220
20% censoring			
$\hat{\gamma}_{\text{AE}}$	0.2377	0.0930	0.2246
$\hat{\gamma}_{\text{SE}}$	0.2603	0.1086	0.2504
$\hat{\gamma}_{\text{LNX(-2)}}$	0.3095	0.1543	0.3009
$\hat{\gamma}_{\text{LNX(2)}}$	0.2249	0.0819	0.2112
$\hat{\gamma}_{\text{GE(-2)}}$	0.2843	0.1261	0.2757
$\hat{\gamma}_{\text{GE(2)}}$	0.1990	0.0687	0.1770

The plots in Figures 4.2 and 4.3 show the results for the CRE distribution with a parameter value of $\gamma = 0.5$. As expected, the non-censored sample leads to better performance and the estimators have accurate results. With 20% censoring, slightly degraded performance due to overestimation is seen, with optimal coverage, MSE and bias at a parameter value of about 0.7. Table 4.1 summarises the MAE, MSE and bias attained for the various estimators. Two arbitrary values of a and k were chosen for the estimators that correspond to the asymmetric loss functions. It is not meaningful to compare their accuracy measures with those of the symmetric loss functions, since the mean errors and bias could be made as small as desired by choosing the values of a and k appropriately.

In Figures 4.4 and 4.5 the results for the CRE distribution with a parameter value of $\gamma = 1.25$ and $n = 50$ are shown. Table 4.2 shows the MAE, MSE and bias of the estimators. Again, good results are seen with no censoring, and overestimation with 20% censoring. In the latter case, the parameter value corresponding to optimal coverage, MSE and bias is about 1.75. A similar set of results, but with $n = 30$, are shown in Figures 4.6 and 4.7 and Table 4.3. It seems that a reduction in sample size lead to slightly decreased accuracy and precision of the estimators.

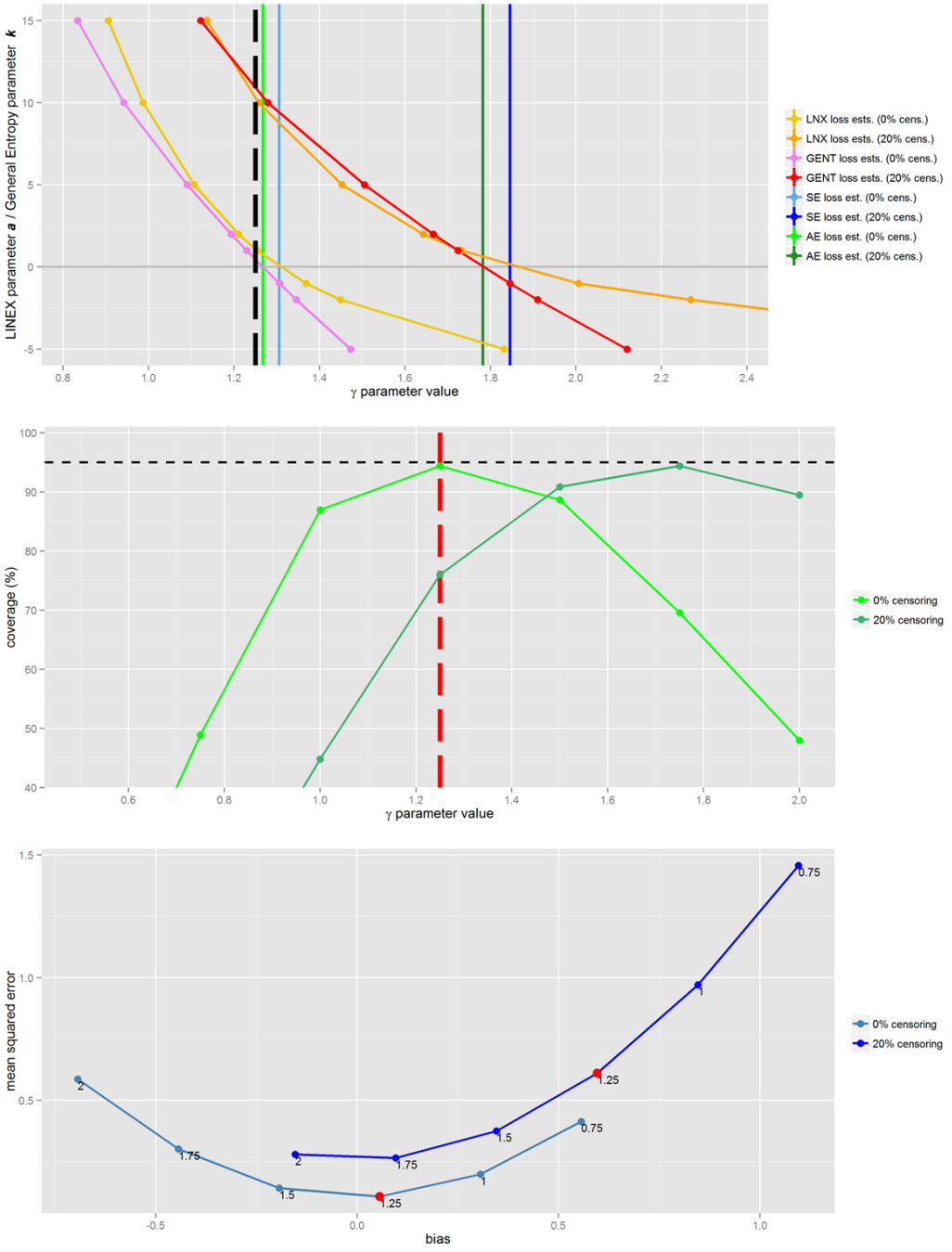


Figure 4.4: *Plots of Bayesian estimates (top), coverage (middle) and MSE against bias (bottom) for CRE model with* $\gamma = 1.25$*, and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours).*

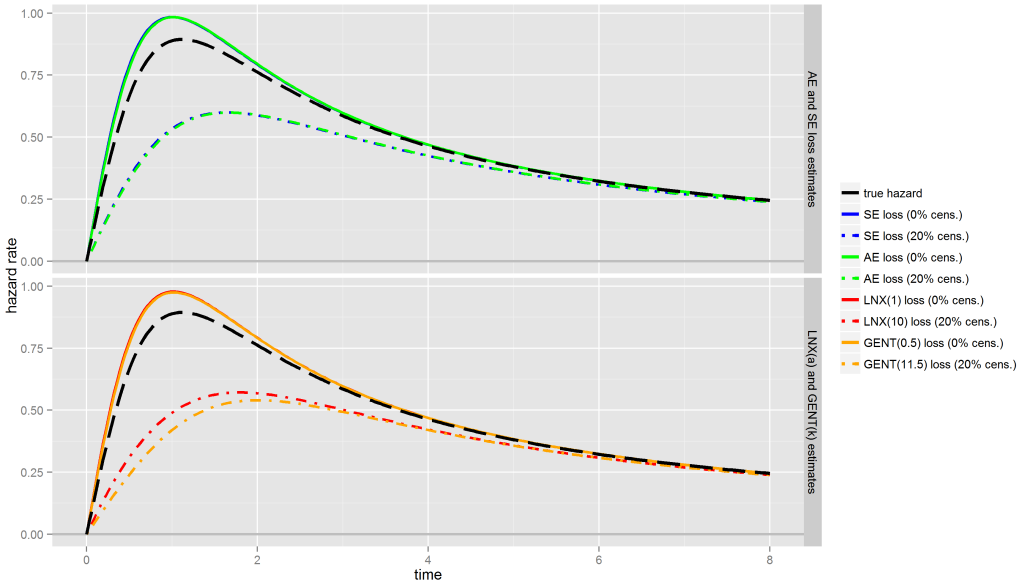


Figure 4.5: *Plots of Bayesian estimates of the hazard function for the CRE model with $\gamma = 1.25$ and two levels of censoring, derived using four different loss functions.*

Table 4.2: *The MAE, MSE and bias of estimators for the CRE model, with parameter $\gamma = 1.25$.*

estimator	MAE	MSE	bias
0% censoring			
$\hat{\gamma}_{\text{AE}}$	0.2433	0.0992	0.0175
$\hat{\gamma}_{\text{SE}}$	0.2496	0.1082	0.0561
$\hat{\gamma}_{\text{LNX(-2)}}$	0.3303	0.2103	0.1974
$\hat{\gamma}_{\text{LNX(2)}}$	0.2260	0.0807	-0.0407
$\hat{\gamma}_{\text{GE(-2)}}$	0.2615	0.1224	0.0947
$\hat{\gamma}_{\text{GE(2)}}$	0.2429	0.0922	-0.0578
20% censoring			
$\hat{\gamma}_{\text{AE}}$	0.5676	0.5204	0.5323
$\hat{\gamma}_{\text{SE}}$	0.6235	0.6110	0.5959
$\hat{\gamma}_{\text{LNX(-2)}}$	1.0323	1.7458	1.0159
$\hat{\gamma}_{\text{LNX(2)}}$	0.4408	0.3164	0.3915
$\hat{\gamma}_{\text{GE(-2)}}$	0.6823	0.7127	0.6584
$\hat{\gamma}_{\text{GE(2)}}$	0.4729	0.3800	0.4147

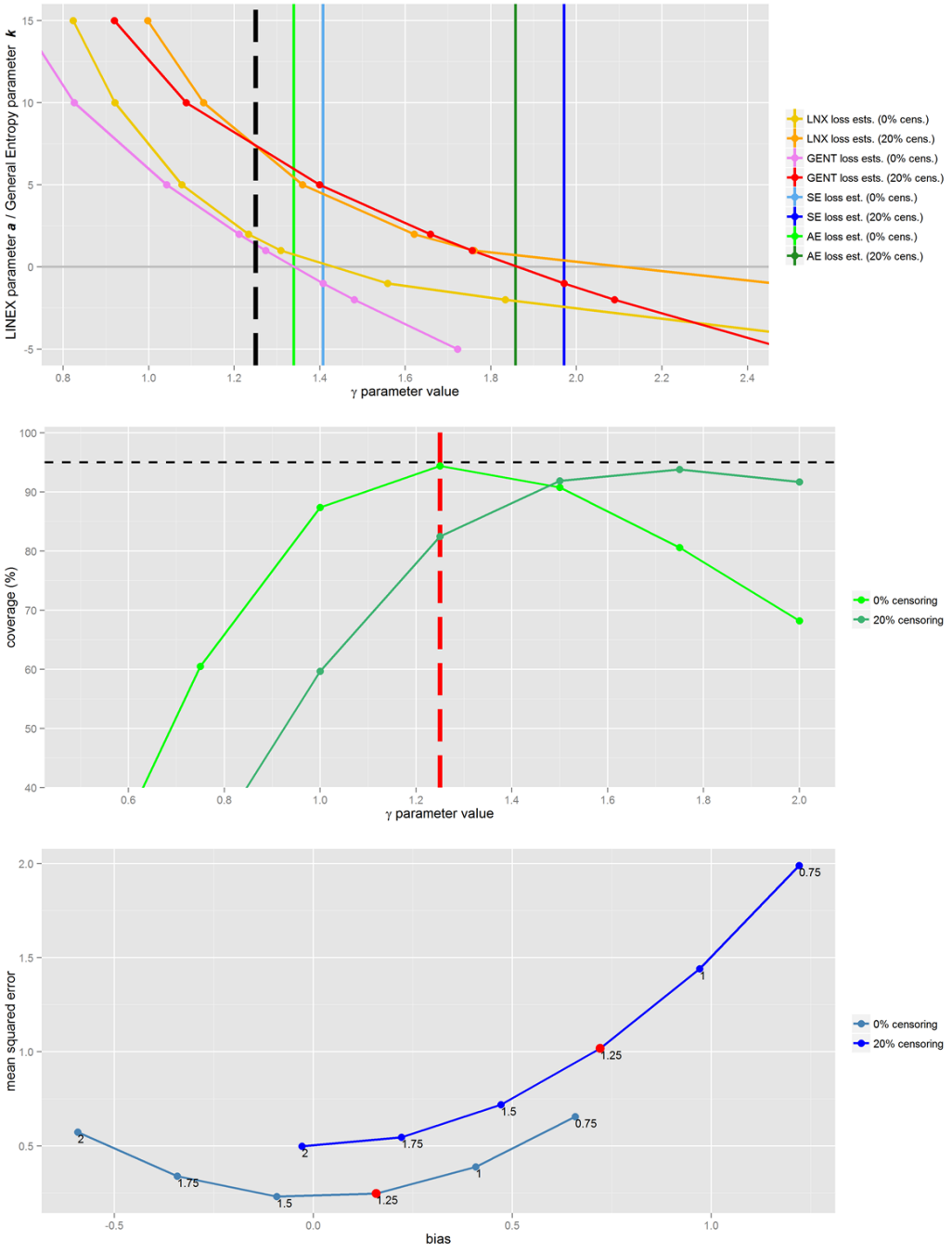


Figure 4.6: *Plots of Bayesian estimates (top), coverage (middle) and MSE against bias (bottom) for CRE model with* $\gamma = 1.25$*, and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours), and with sample size* $n = 30$*.*

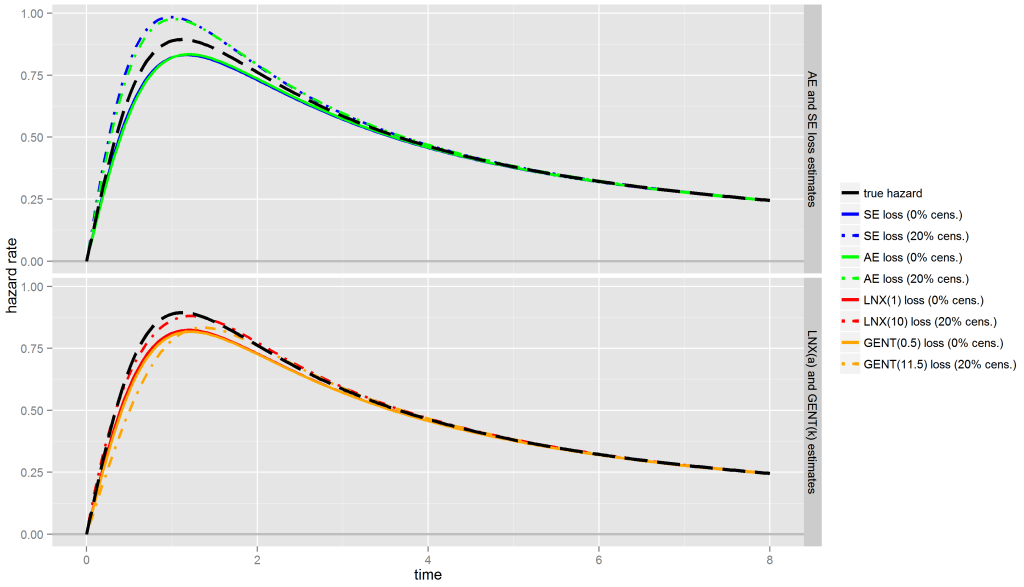


Figure 4.7: *Plots of Bayesian estimates of the hazard function for the CRE model with* $\gamma = 1.25$*, and two levels of censoring, derived using four different loss functions and sample size* $n = 30$*.*

Table 4.3: *The MAE, MSE and bias of estimators for the CRE model, with parameter* $\gamma = 1.25$ *and* $n = 30$*.*

estimator	MAE	MSE	bias
0% censoring			
$\hat{\gamma}_{\text{AE}}$	0.3505	0.2096	0.0891
$\hat{\gamma}_{\text{SE}}$	0.3744	0.2478	0.1581
$\hat{\gamma}_{\text{LNX(-2)}}$	0.7132	1.1447	0.5816
$\hat{\gamma}_{\text{LNX(2)}}$	0.2957	0.1356	-0.0174
$\hat{\gamma}_{\text{GE(-2)}}$	0.4074	0.3010	0.2291
$\hat{\gamma}_{\text{GE(2)}}$	0.3282	0.1677	-0.0398
20% censoring			
$\hat{\gamma}_{\text{AE}}$	0.6769	0.8054	0.6081
$\hat{\gamma}_{\text{SE}}$	0.7735	1.0176	0.7212
$\hat{\gamma}_{\text{LNX(-2)}}$	2.1129	8.9876	2.0867
$\hat{\gamma}_{\text{LNX(2)}}$	0.4632	0.3655	0.3691
$\hat{\gamma}_{\text{GE(-2)}}$	0.8785	1.2724	0.8372
$\hat{\gamma}_{\text{GE(2)}}$	0.5245	0.5067	0.4074

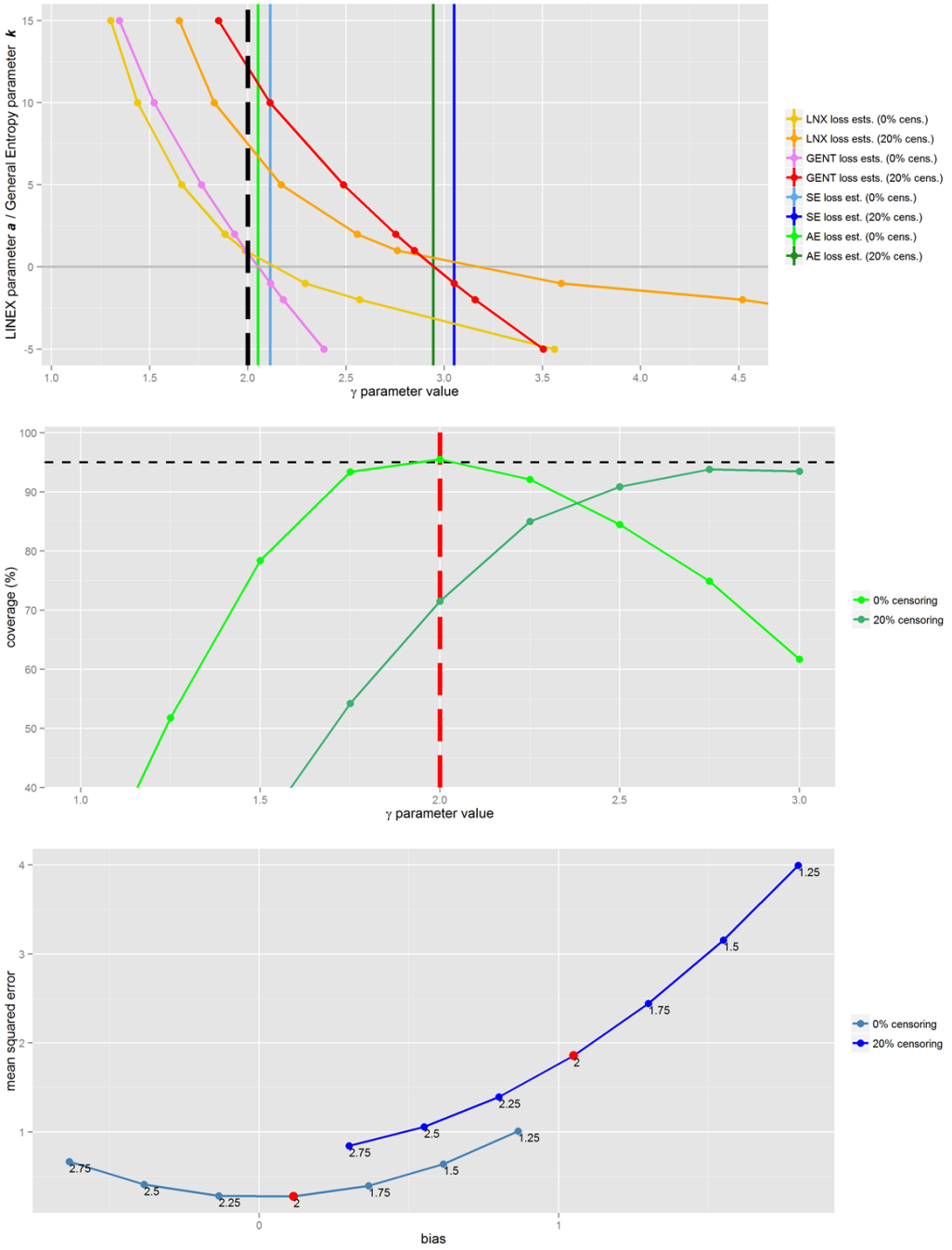


Figure 4.8: *Plots of Bayesian estimates (top), coverage (middle) and MSE against bias (bottom) for CRE model with* $\gamma = 2$*, and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours).*

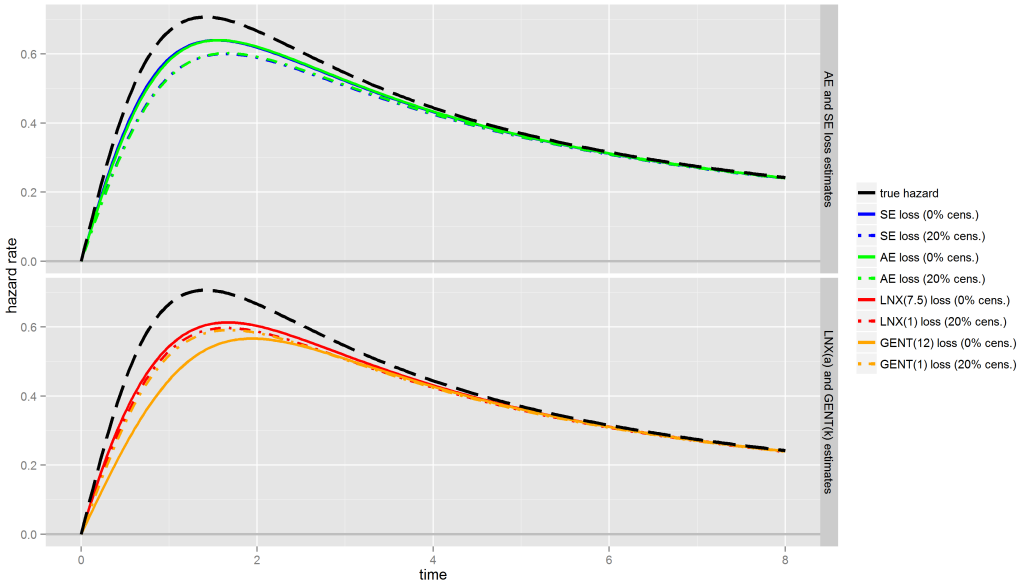


Figure 4.9: *Plots of Bayesian estimates of the hazard function for the CRE model with $\gamma = 2$ and two levels of censoring, derived using four different loss functions.*

Table 4.4: *The MAE, MSE and bias of estimators for the CRE model, with parameter $\gamma = 2$.*

estimator	MAE	MSE	bias
0% censoring			
$\hat{\gamma}_{\text{AE}}$	0.3909	0.2481	0.0514
$\hat{\gamma}_{\text{SE}}$	0.4071	0.2748	0.1148
$\hat{\gamma}_{\text{LNX(-2)}}$	0.7226	0.9622	0.5661
$\hat{\gamma}_{\text{LNX(2)}}$	0.3498	0.1851	-0.1187
$\hat{\gamma}_{\text{GE(-2)}}$	0.4313	0.3143	0.1776
$\hat{\gamma}_{\text{GE(2)}}$	0.3826	0.2260	-0.0704
20% censoring			
$\hat{\gamma}_{\text{AE}}$	0.9873	1.5885	0.9445
$\hat{\gamma}_{\text{SE}}$	1.0817	1.8561	1.0502
$\hat{\gamma}_{\text{LNX(-2)}}$	2.5256	10.3824	2.5132
$\hat{\gamma}_{\text{LNX(2)}}$	0.6415	0.6883	0.5548
$\hat{\gamma}_{\text{GE(-2)}}$	1.1814	2.1540	1.1542
$\hat{\gamma}_{\text{GE(2)}}$	0.8266	1.1702	0.7495

Figures 4.8 and 4.9 show the results for the CRE distribution with a parameter value of $\gamma = 2$. Similar to the previous cases, accurate estimates were obtained with no censoring, while 20% censoring lead to overestimation, for which a parameter value of about 3 seemed to give optimal results. The estimators' mean errors and bias are summarised in Table 4.4.

4.1.5 Discussion

In this section, results of the simulation study for the CRE model, showcased in the previous section, are briefly discussed.

Across the results of all parameter configurations, censoring caused overestimation of the true value of γ. Without censoring, good frequentist properties were observed, as 95% of the 1000 credible intervals for γ included the true value of 0.5. It is also clear to see that with no censoring, the lowest MSE and bias are obtained for the true parameter value. However, with 20% censoring, the coverage was only 75% and both the MSE and bias were increased for all parameter values. Due to the overestimation, the optimal coverage, MSE and bias was observed at a higher parameter value of about 1.4 times the true value.

For one choice of parameter value, the simulation was repeated with a decreased sample size. As expected, this lead to a slightly less accurate estimators with higher variances.

Between the two symmetric loss estimators, $\hat{\gamma}_{\text{SE}}$ and $\hat{\gamma}_{\text{AE}}$, the latter emerged as the most accurate, even though they do not differ by much. It can be seen that $\hat{h}_{\text{AE}}(t, \gamma)$ and $\hat{h}_{\text{SE}}(t, \gamma)$ are also very similar. In all cases, though, $\hat{\gamma}_{\text{AE}}$ had lower measures of accuracy (MAE, MSE and bias).

The two asymmetric loss functions, LINEX and GE loss, each has their own parameter controlling the degree of asymmetry. Both of their estimators exhibit a monotonically decreasing curve for varying values of their parameters, but in all cases the trend of $\hat{\gamma}_{\text{GE(k)}}$ seems closer to linearity than that of $\hat{\gamma}_{\text{LNX(a)}}$. Lastly, the values of the loss parameters that correspond to the true γ value are considered. For GE loss, one can see that in all cases, a value of about $k = 1$ corresponds to the true γ with no censoring, while a value of about $k = 12$ corresponds to the true value in the censored case. However, censoring

seems to have a much larger effect on the LINEX loss estimators, and the general range for the a parameter that yields a true γ value changes quite drastically as γ varies.

In general, the shape of the true hazard rate seems to be estimated fairly well, especially in the non-censored case. In the following section, a compound Rayleigh model with an additional scale parameter is investigated in a simulation study.

4.2 The CRG model

4.2.1 Model characteristics

In Section 3.2.2.2, the Rayleigh is compounded with respect to a Gamma distribution, resulting in the CRG model with PDF and CDF in equations (3.7) and (3.8) respectively. These are used to find the survival function (2.1)

$$S(t, \alpha, \beta) = \beta^{\alpha}(t^2 + \beta)^{-\alpha},$$

from which the hazard rate (2.2) can also be derived as follows

$$h(t, \alpha, \beta) = \frac{2\alpha t}{t^2 + \beta}. \tag{4.8}$$

The stationary point is at $t = \sqrt{\beta}$, allowing it to model a variety of scenarios depending on the model parameters. The form of this hazard function is presented in Figure 4.10.

The elements of the Fisher information matrix (2.11) are expected values of second-order partial derivatives of the logarithm of (3.7). Using the definition

$$l_f = \ln f(t|\alpha, \beta) = \ln 2 + \ln \alpha + \alpha \ln \beta + \ln t - (\alpha + 1)\ln(t^2 + \beta),$$

the following second-order derivatives can be calculated

$$\begin{aligned}
\frac{\partial^2 l_f}{\partial \alpha^2} &= -\frac{1}{\alpha^2} \\
\frac{\partial^2 l_f}{\partial \alpha \partial \beta} &= \frac{1}{\beta} - \frac{1}{t^2 + \beta} \\
\frac{\partial^2 l_f}{\partial \beta^2} &= -\frac{\alpha}{\beta^2} + \frac{\alpha + 1}{(t^2 + \beta)^2}.
\end{aligned}$$

Consequently, the Fisher information $\mathcal{I}_F$ can be formulated as

$$\mathcal{I}_F = \mathrm{E}_{T|\alpha,\beta}\begin{bmatrix} \frac{1}{\alpha^2} & \frac{1}{t^2+\beta} - \frac{1}{\beta} \\ \frac{1}{t^2+\beta} - \frac{1}{\beta} & \frac{\alpha}{\beta^2} - \frac{\alpha+1}{(t^2+\beta)^2} \end{bmatrix}.$$

It is clear that two expected values with respect to the CRG distribution are now required, $\mathrm{E}\left[\frac{1}{T^2+\beta}\right]$ and $\mathrm{E}\left[\frac{1}{(T^2+\beta)^2}\right]$. In order to find them, the integrands are rewritten in a form that is comparable to the PDF of the Beta Prime distribution (see Appendix A.2). Note that for the first expected value

$$\begin{aligned}\mathrm{E}\left[\frac{1}{T^2+\beta}\right] &= \int_0^\infty 2\alpha\beta^\alpha t(t^2+\beta)^{-\alpha-2}dt \\ &= \alpha\beta^{-\frac{3}{2}}\int_0^\infty 2\left(\frac{t}{\sqrt{\beta}}\right)\left[1+\left(\frac{t}{\sqrt{\beta}}\right)^2\right]^{-\alpha-2} dt.\end{aligned}$$

This integral can be solved by relating it to a Beta Prime distribution with parameters $p \equiv 2$, $q \equiv \sqrt{\beta}$, $r \equiv 1$, and $s \equiv \alpha + 1$. Thus

$$\begin{aligned}\mathrm{E}\left[\frac{1}{T^2+\beta}\right] &= \alpha\beta^{-\frac{3}{2}}\sqrt{\beta}B(1,\alpha+1) \\ &= \frac{\alpha}{\beta(\alpha+1)},\end{aligned}$$

after simplification of the Beta function. The second expected value has the form

$$\mathrm{E}\left[\frac{1}{(T^2+\beta)^2}\right] = \int_0^\infty 2\alpha\beta^\alpha t(t^2+\beta)^{-\alpha-3}dt$$

and similar to what was done previously, the integral can be reformulated to the form of a Beta Prime distribution with parameters $p \equiv 2$, $q \equiv \sqrt{\beta}$, $r \equiv 1$, but $s \equiv \alpha + 2$. The solution now becomes

$$\begin{aligned}\mathrm{E}\left[\frac{1}{(T^2+\beta)^2}\right] &= \alpha\beta^{-\frac{5}{2}}\sqrt{\beta}B(1,\alpha+2) \\ &= \frac{\alpha}{\beta^2(\alpha+2)}\end{aligned}$$

after simplification.

Consequently, using the results above as well as the fact that the expected value over a term independent of t falls away, the simplified form of the Fisher information matrix is

given by

$$\begin{aligned}\mathcal{I}_F &= \begin{bmatrix} \frac{1}{\alpha^2} & \mathrm{E}[\frac{1}{T^2+\beta}] - \frac{1}{\beta} \\ \mathrm{E}[\frac{1}{T^2+\beta}] - \frac{1}{\beta} & \frac{\alpha}{\beta^2} - \mathrm{E}\left[\frac{\alpha+1}{(T^2+\beta)^2}\right] \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{\alpha^2} & \frac{\alpha}{\beta(\alpha+1)} - \frac{1}{\beta} \\ \frac{\alpha}{\beta(\alpha+1)} - \frac{1}{\beta} & \frac{\alpha}{\beta^2} - \frac{\alpha(\alpha+1)}{\beta^2(\alpha+2)} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{\alpha^2} & \frac{-1}{\beta(\alpha+1)} \\ \frac{-1}{\beta(\alpha+1)} & \frac{\alpha}{\beta^2(\alpha+2)} \end{bmatrix}. \end{aligned} \tag{4.9}$$

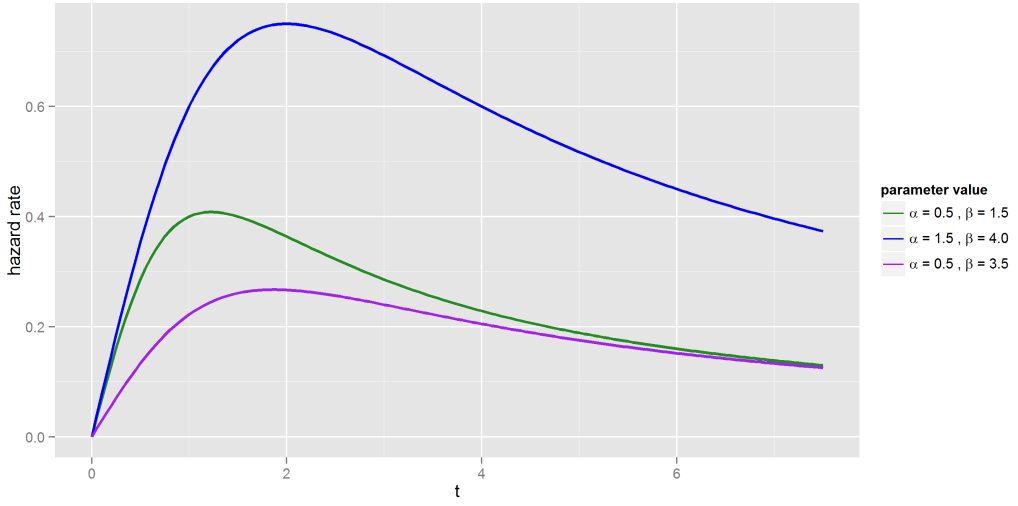


Figure 4.10: *Hazard rate of CRG model for various values of its parameters.*

4.2.2 Prior and posterior distributions

In order to derive the posterior distribution for the parameters of the CRG model, prior distributions first need to be constructed. Here, there is scope to investigate objective prior distributions beyond just the Jeffreys prior, since there is more than one model parameter. In addition, two reference priors, using both orderings of the parameters, as well as the PM prior are considered. This section focuses on the derivation of these prior distributions specifically in the case of the CRG model.

4.2.2.1 Derivation of the Jeffreys prior

When there are multiple model parameters, Jeffreys' prior is defined to be proportional to the square root of the determinant of the Fisher information matrix $\mathcal{I}_F$.

The determinant of (4.9) is

$$\begin{aligned} |\mathcal{I}_F| &= \frac{\alpha}{\alpha^2\beta^2(\alpha+2)} - \frac{1}{\beta^2(\alpha+1)^2} \\ &= \frac{1}{\alpha\beta^2(\alpha+2)(\alpha+1)^2}, \end{aligned} \tag{4.10}$$

thus the proportional form of Jeffreys' prior is given by

$$\pi_{\text{jeff}}(\alpha,\beta) \propto \frac{1}{\beta(\alpha+1)\sqrt{\alpha(\alpha+2)}}. \tag{4.11}$$

4.2.2.2 Derivation of the reference priors

In Section 2.2.4.2, the reasoning and derivation of the reference prior is discussed. This popular non-informative approach stems from maximising the Kullback-Leibler divergence between prior and posterior. As discussed, the one-at-a-time rule will be used by considering both orderings $\{\alpha,\beta\}$ and $\{\beta,\alpha\}$, ultimately obtaining two distinct reference prior distributions.

Initially, the parameter β is considered as the primary variable. The first step of the reference prior algorithm is to choose a conditional prior for α. A natural choice is the Jeffreys prior, which can easily be found from the top left element of the Fisher information matrix (4.9), such that

$$\pi(\alpha|\beta) \propto \sqrt{\frac{1}{\alpha^2}} = \frac{1}{\alpha}.$$

Since this prior is improper, the next step is to consider a sequence $\Theta_1 \subset \Theta_2 \subset \ldots$ of subsets such that their union is the parameter space of (α,β) and the density $\pi(\alpha|\beta)$ has finite mass on $\Omega_{i,\beta} = \{\alpha : (\alpha,\beta) \in \Theta_i\}$ for all β. One option for these subsets amounts to choosing rectangles on the two-dimensional parameter space, bounded between a_{1i} and a_{2i} on the axis for α, and between b_{1i} and b_{2i} on the β-axis. Then, as $i \to \infty$, these rectangles become bigger and bigger, i.e. a_{1i} and b_{1i} tend towards 0, while a_{2i} and b_{2i} tend towards infinity. In this way, for all β, the conditional prior is finite on $\Omega_{i,\beta}$. The

normalised conditional densities (2.14) now become

$$\begin{aligned}\pi_i(\alpha|\beta) &= K_i(\beta)\pi(\alpha|\beta)\\ &= \frac{\pi(\alpha|\beta)}{\int_{\Omega_{i,\beta}} \pi(\alpha|\beta)d\beta}\\ &= \frac{1}{\alpha(\ln a_{2i} - \ln a_{1i})}.\end{aligned}$$

The final two steps involve finding the marginal distribution of β as in (2.15) and thereafter the joint distribution – i.e. the reference prior. For each $\Omega_{i,\beta}$, it is true that

$$\begin{aligned}\pi_i(\beta) &= \exp\left\{\frac{1}{2}\int_{\Omega_{i,\beta}} \pi(\alpha|\beta)\ln\left(\frac{|\mathcal{I}_F|}{[\mathcal{I}_F]_{22}}\right)d\alpha\right\}\\ &= \exp\left\{\frac{1}{2}\int_{a_{1i}}^{a_{2i}} \frac{1}{\alpha(\ln a_{2i} - \ln a_{1i})}\ln\left(\frac{\alpha^2}{\alpha\beta^2(\alpha+2)(\alpha+1)^2}\right)d\alpha\right\}\\ &= \exp\left\{\frac{1}{\alpha a_1'}\left(\ln\alpha - 2\ln\beta - 2\ln(\alpha+1) - \ln(\alpha+2)\right)d\alpha\right\}\\ &= a_2'\exp(-\ln\beta)\\ &\propto \frac{1}{\beta}\end{aligned}$$

where the determinant of the Fisher information in the second line was derived previously in (4.10), and a_1' and a_2' are substitutions for nuisance terms not relevant to the concluding proportionality. In the last step, the reference prior with β-primary ordering can be found as in equation (2.16). Now, a fixed point β^* needs to be chosen from the parameter space of β. Suppose $\beta^* = 1$, then, since

$$K_i(\beta) = K_i(\beta^*) = \frac{1}{\ln a_{2i} - \ln a_{1i}}$$

it follows that

$$\begin{aligned}\pi(\alpha,\beta)^{\beta}_{\text{ref}} &= \lim_{i\to\infty}\frac{\frac{1}{\beta}K_i(\beta)}{K_i(1)}\pi(\alpha|\beta)\\ &\propto \frac{1}{\alpha\beta}.\end{aligned} \tag{4.12}$$

The steps discussed above can be used in exactly the same manner, albeit with small differences, to yield the reference prior with the α-primary ordering of the parameters. To this end, a conditional prior for β is considered, the Jeffreys prior using the bottom

right element of (4.9), such that

$$\pi(\beta|\alpha) \propto \sqrt{\frac{\alpha}{\beta^2(\alpha+2)}}.$$

This is improper and the exact same reasoning as before can be used, by choosing the subsets $\Theta_1 \subset \Theta_2 \subset \ldots$ as rectangles in the parameter space, such that $\pi(\beta|\alpha)$ has finite mass on $\Omega_{i,\alpha} = \{\beta : (\alpha, \beta) \in \Theta_i\}$ for all α. Now,

$$\begin{aligned} K_i(\alpha) &= \left(\int_{b_{1i}}^{b_{2i}} \sqrt{\frac{\alpha}{\beta^2(\alpha+2)}} d\beta \right)^{-1} \\ &= \sqrt{\frac{\alpha+2}{\alpha}} \left(\frac{1}{\ln b_{2i} - \ln b_{1i}} \right) \end{aligned}$$

and with this, the normalised densities become

$$\pi_i(\beta|\alpha) = \frac{1}{\beta(\ln b_{2i} - \ln b_{1i})}.$$

At this point, the marginal density of α with respect to i is required, but only in its proportional form. Note that from (4.10),

$$\ln\left(\frac{|\mathcal{I}_F|}{[\mathcal{I}_F]_{22}}\right) = -2\ln\alpha - 2\ln(\alpha+1),$$

thus

$$\begin{aligned} \pi_i(\alpha) &= \exp\left\{ \frac{1}{2} \int_{\Omega_{i,\alpha}} \pi(\beta|\alpha) \ln\left(\frac{|\mathcal{I}_F|}{[\mathcal{I}_F]_{22}}\right) d\beta \right\} \\ &= \exp\left\{ \frac{1}{2} \int_{b_{1i}}^{b_{2i}} \frac{-2\ln\alpha - 2\ln(\alpha+1)}{\beta(\ln b_{2i} - \ln b_{1i})} d\beta \right\} \\ &\propto \exp\left\{-2\ln\alpha - 2\ln(\alpha+1)\right\} \\ &\propto \frac{1}{\alpha(\alpha+1)}. \end{aligned}$$

The last step entails deriving the reference prior according to (2.16). A fixed point $\alpha^* = 1$ is chosen, leading to

$$\begin{aligned} K_i(\alpha^*) &= \frac{\sqrt{2}}{\ln b_{2i} - \ln b_{1i}} \\ \text{and } \pi_i(\alpha^*) &= \frac{1}{2}, \end{aligned}$$

using the solutions for $K_i(\alpha)$ and $\pi_i(\alpha)$ found above. Furthermore, using the conditional density for β, the joint density becomes

$$\pi(\alpha,\beta) = \lim_{i\to\infty} \frac{\frac{1}{\alpha(\alpha+1)}\left(\sqrt{\frac{\alpha+2}{\alpha}}\right)\left(\frac{1}{\ln b_{2i}-\ln b_{1i}}\right)}{\frac{\sqrt{2}}{2}\left(\frac{1}{\ln b_{2i}-\ln b_{1i}}\right)} \cdot \sqrt{\frac{\alpha}{\beta^2(\alpha+2)}}.$$

Finally, simplifying the equation above and omitting nuisance terms, the α-primary ordering reference prior distribution emerges in proportional form as

$$\pi(\alpha,\beta)^{\alpha}_{\text{ref}} \propto \frac{1}{\alpha\beta(\alpha+1)}. \tag{4.13}$$

4.2.2.3 Derivation of the probability matching prior

The PM prior attempts to equate probabilities given by frequentist confidence intervals and Bayesian credible intervals (Section 2.2.4.3). To satisfy this condition, summarised in equation (2.17), a prior $\pi(\alpha,\beta)$ needs to be derived such that (2.18) is satisfied.

First, the inverse of the Fisher information matrix is required. From (4.9), this can be shown to be

$$\mathcal{I}_F^{-1} = \begin{bmatrix} \alpha^2(\alpha+1)^2 & \beta\alpha(\alpha+1)(\alpha+2) \\ \beta\alpha(\alpha+1)(\alpha+2) & \frac{\beta^2}{\alpha}(\alpha+1)^2(\alpha+2) \end{bmatrix}.$$

The condition (2.18) to be satisfied now amounts to the following

$$\frac{\partial}{\partial\alpha}\left\{\pi(\alpha,\beta)\left([\mathcal{I}_F^{-1}]_{1,1}\right)^{\frac{1}{2}}\right\} + \frac{\partial}{\partial\beta}\left\{\pi(\alpha,\beta)[\mathcal{I}_F^{-1}]_{1,2}\left([\mathcal{I}_F^{-1}]_{1,1}\right)^{-\frac{1}{2}}\right\} = 0.$$

Using the relevant elements of $\mathcal{I}_F^{-1}$, this can be simplified to

$$\frac{\partial}{\partial\alpha}\{\pi(\alpha,\beta)\alpha(\alpha+1)\} + \frac{\partial}{\partial\beta}\{\pi(\alpha,\beta)\beta(\alpha+2)\} = 0. \tag{4.14}$$

A simple argument is used to find the proportional form of the prior distribution. If $\pi(\alpha,\beta) \propto \{\alpha(\alpha+1)\}^{-1}$, then

$$\frac{\partial}{\partial\alpha}\{\pi(\alpha,\beta)\alpha(\alpha+1)\} = 0.$$

Similarly, if $\pi(\alpha, \beta) \propto \beta^{-1}$, then

$$\frac{\partial}{\partial \beta} \{\pi(\alpha, \beta)\beta(\alpha + 2)\} = 0.$$

Consequently, these two statements yield the PM prior, since by assigning

$$\pi_{\text{PM}}(\alpha, \beta) \propto \frac{1}{\alpha\beta(\alpha + 1)}, \tag{4.15}$$

the condition (4.14) and thus (2.18) is satisfied.

4.2.2.4 Derivation of posterior distributions

It is interesting, if quite reassuring, to note that one of the references priors (4.13) and the PM prior (4.15) are equivalent. For this reason, the reference prior with β-primary ordering (4.12) shall henceforth be denoted as only $\pi_{\text{ref}}(\alpha, \beta)$.

Consider a sample of n survival times $\mathbf{t} = (t_1, t_2, \ldots, t_n)$, ordered such that the first d are non-censored and the remaining $(n - d)$ right censored, the likelihood function now follows as

$$\begin{aligned}
\mathcal{L}(\alpha, \beta|\mathbf{t}) &\propto \prod_{i=1}^{d} f(t_i|\alpha, \beta) \prod_{j=d+1}^{n} S(t_j, \alpha, \beta) \\
&\propto \prod_{i=1}^{d} \frac{2t_i\alpha\beta^{\alpha}}{(t_i^2 + \beta)^{\alpha+1}} \prod_{j=d+1}^{n} \frac{\beta^{\alpha}}{(t_j^2 + \beta)^{\alpha}} \\
&\propto (2\alpha)^d \prod_{i=1}^{d} \frac{t_i}{(t_i^2 + \beta)} \prod_{j=1}^{n} \left(1 + \frac{t_j^2}{\beta}\right)^{-\alpha}.
\end{aligned}$$

For computational reasons, some of the terms above are reformulated in terms of logs, such that the likelihood function becomes

$$\begin{aligned}
\mathcal{L}(\alpha, \beta|\mathbf{t}) &\propto (2\alpha)^d e^{W_1(\beta) - \alpha W_2(\beta)} \\
\text{where } W_1(\beta) &= \sum_{i=1}^{d} \ln\left(\frac{t_i}{t_i^2 + \beta}\right) \\
\text{and } W_2(\beta) &= \sum_{i=1}^{n} \ln\left(1 + \frac{t_i^2}{\beta}\right)
\end{aligned}$$

The proportional form of the posterior distributions corresponding to the different priors can now be constructed by respectively multiplying the Jeffreys prior (4.11), the reference prior (4.12) and the PM prior (4.15) with the likelihood function $\mathcal{L}(\alpha, \beta|\mathbf{t})$ above.

4.2.3 Bayesian estimators of the parameters

The posterior distribution of the parameters as well as a loss function are required to formally defined the Bayesian estimators of the CRG model parameters. In this section, the forms of the estimators under the symmetric and asymmetric loss functions are given with regards to an arbitrary posterior distribution of α and β. In the simulation study, each of these estimators will be computed using all three posterior distributions corresponding to the different priors, i.e. the Jeffreys (4.11), reference (4.12) and PM (4.15) priors inferred in the previous section.

Under the symmetric AE and SE loss functions, the estimators become the posterior expected values and medians

$$\begin{aligned}\left(\hat{\alpha}_{\mathrm{AE}},\ \hat{\beta}_{\mathrm{AE}}\right) &= \mathrm{median}_{\alpha,\beta|\mathbf{t}}[(\alpha,\beta)] \\ \left(\hat{\alpha}_{\mathrm{SE}},\ \hat{\beta}_{\mathrm{SE}}\right) &= \mathrm{E}_{\alpha,\beta|\mathbf{t}}[(\alpha,\beta)].\end{aligned}$$

and using the asymmetric LINEX loss function with its parameter a, and the GE loss function with its parameter k, the Bayesian estimators are given by

$$\begin{aligned}\left(\hat{\alpha}_{\mathrm{LNX(a)}},\ \hat{\beta}_{\mathrm{LNX(a)}}\right) &= -\frac{1}{a}\ln \mathrm{E}_{\alpha,\beta|\mathbf{t}}\left[e^{-a(\alpha,\beta)}\right] \\ \left(\hat{\alpha}_{\mathrm{GE(k)}},\ \hat{\beta}_{\mathrm{GE(k)}}\right) &= \left(\mathrm{E}_{\alpha,\beta|\mathbf{t}}\left[(\alpha,\beta)^{-k}\right]\right)^{-\frac{1}{k}}.\end{aligned}$$

In the formulation above, (α, β) denotes the two-dimensional parameter vector.

The estimators under different loss functions for the survival function and hazard rate are calculated by using the parameter estimators defined above in (4.8). In the simulation

study, only the hazard rate estimates are shown and from (4.8), they are defined as

$$\begin{aligned}
\hat{h}_{\mathrm{AE}}(t,\alpha,\beta) &= \mathrm{median}_{\alpha,\beta|\mathbf{t}}[h(t,\alpha,\beta)] \\
\hat{h}_{\mathrm{SE}}(t,\alpha,\beta) &= \mathrm{E}_{\alpha,\beta|\mathbf{t}}[h(t,\alpha,\beta)] \\
\hat{h}_{\mathrm{LNX(a)}}(t,\alpha,\beta) &= -\frac{1}{a}\ln \mathrm{E}_{\alpha,\beta|\mathbf{t}}\left[e^{-ah(t,\alpha,\beta)}\right] \\
\hat{h}_{\mathrm{GE(k)}}(t,\alpha,\beta) &= \left(\mathrm{E}_{\alpha,\beta|\mathbf{t}}\left[h(t,\alpha,\beta)^{-k}\right]\right)^{-\frac{1}{k}}.
\end{aligned}$$

4.2.4 Simulation results

Following the procedures described in Section 3.3.2, a simulation study was carried out to assess the performance of the model parameters' Bayesian estimators, derived in the previous section. This process was performed for three different pairs of parameter values:

$$\begin{aligned}
\alpha &= 0.5\ , \qquad \beta = 1.5 \\
\alpha &= 1.5\ , \qquad \beta = 4.0 \\
\alpha &= 0.5\ , \qquad \beta = 3.5
\end{aligned}$$

and two levels of censoring: $\delta = \{1, 0.8\}$. For illustrative purposes, one of these parameter configurations, $(\alpha = 0.5, \beta = 3.5)$, was studied using two sample sizes, $n = 30$ and $n = 50$. The rest of the simulations used a sample size of 50 throughout.

Each simulation procedure was repeated not only for each parameter configuration and level of censoring, but also for each of the three unique sets of posterior distributions, corresponding to the three non-informative priors derived for the CRG model.

The simulation results are presented graphically in the same way as for the CRE model's results. The exposition of plots are explained in Section 4.1.4. The plots consist of Bayesian point estimates and coverages (Figures 4.11, 4.14, 4.17 and 4.20), MSE against bias (Figures 4.12, 4.15, 4.18 and 4.21) as well as hazard rate estimates (Figures 4.13, 4.16, 4.19 and 4.22). In addition, Tables 4.5 to 4.8 summarise the accuracy measures for the different parameter configurations.

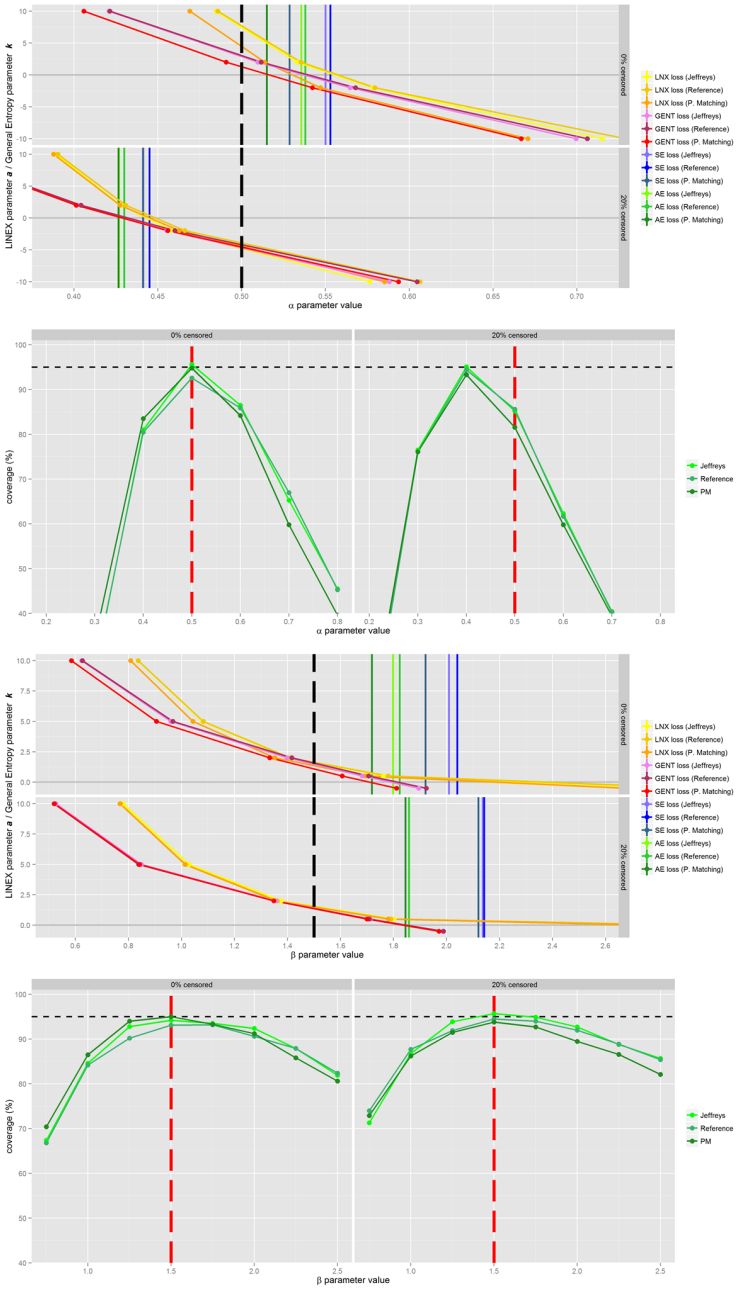


Figure 4.11: *Bayesian estimates and coverages plots for CRG model, with* $\alpha = 0.5$ *(top two) and* $\beta = 1.5$ *(bottom two), and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours).*

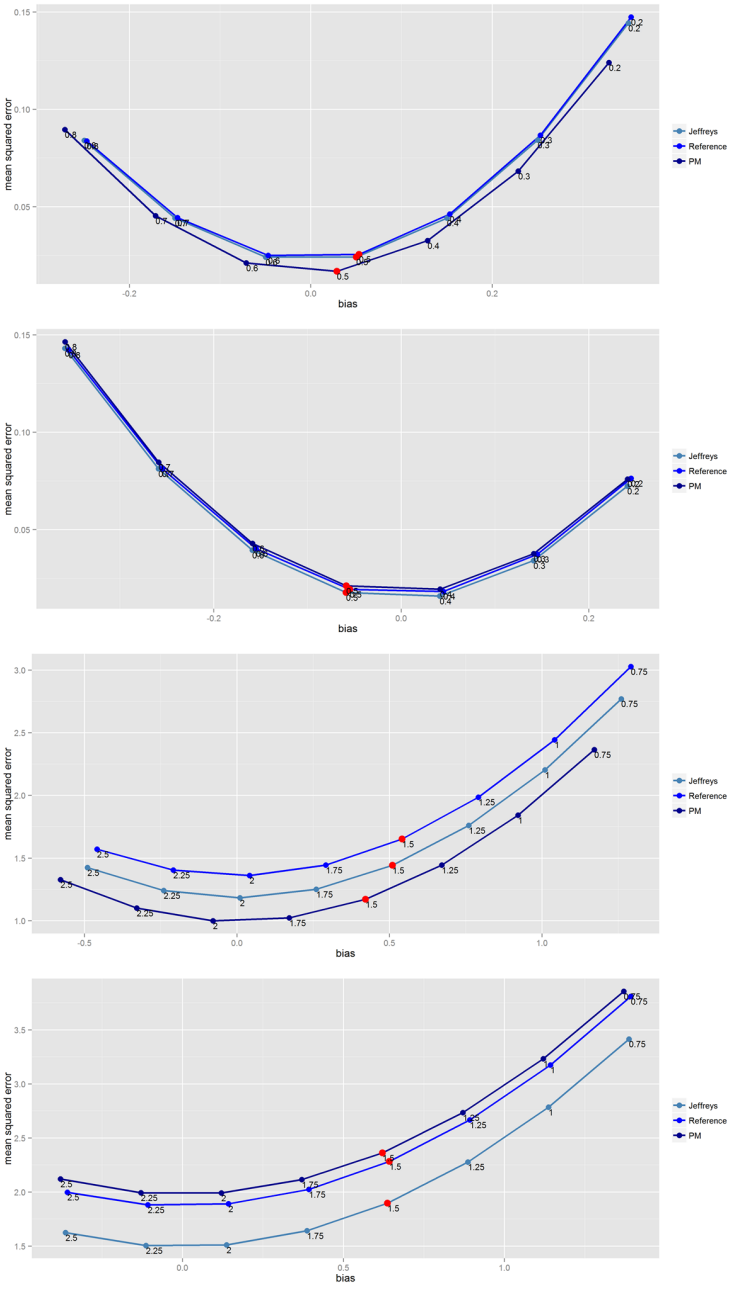


Figure 4.12: *MSE vs bias plots for CRG model, with* $\alpha = 0.5$ *(top two) and* $\beta = 1.5$ *(bottom two), and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours).*

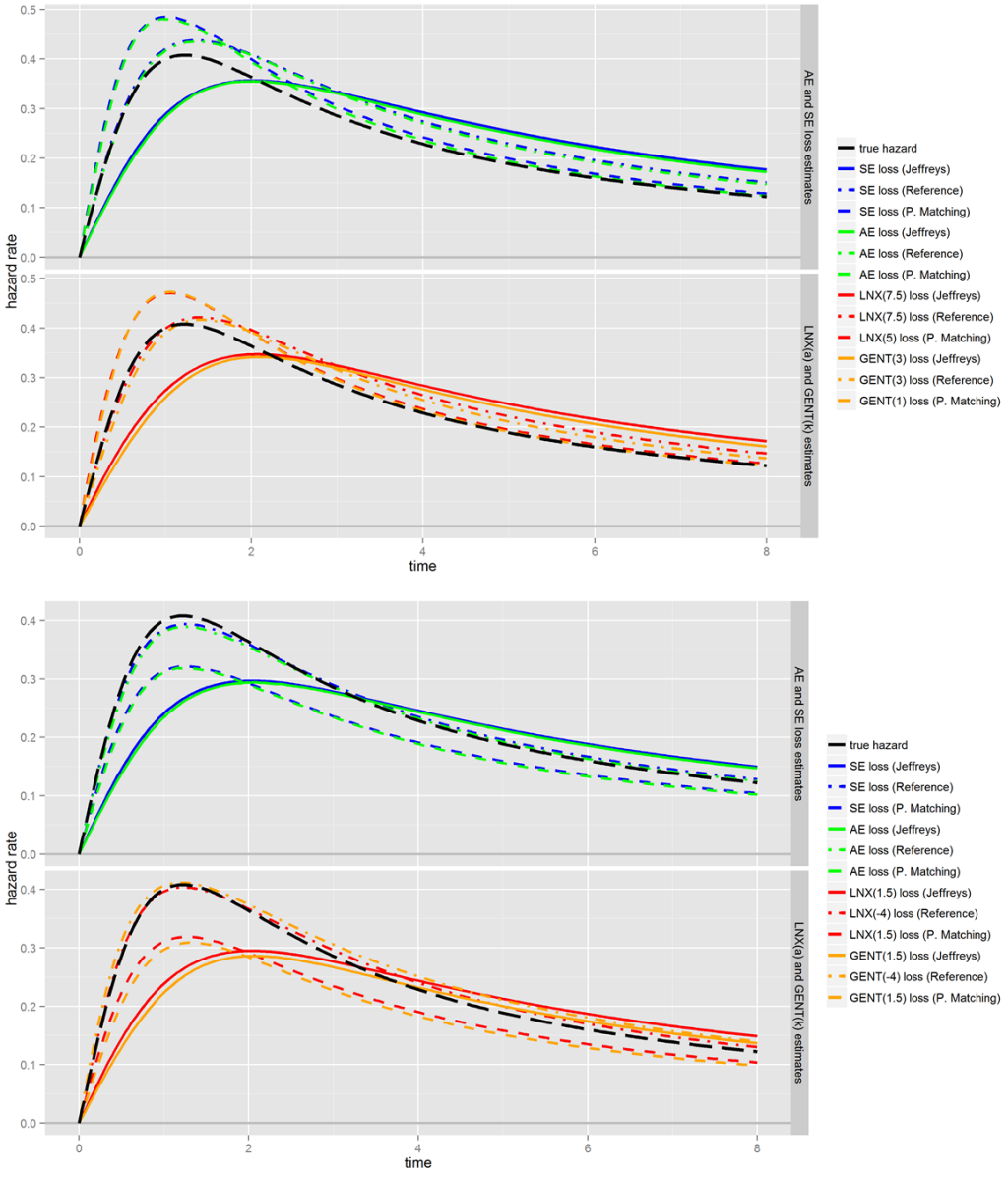


Figure 4.13: *Plots of Bayesian estimates of the hazard function for the CRG model with* $(\alpha, \beta) = (0.5, 1.5)$ *and two levels of censoring (no censoring, top, and 20% censoring, bottom), derived using three different priors and four different loss functions.*

Table 4.5: *The MAE, MSE and bias of estimators for the CRG model, with parameters* $(\alpha, \beta) = (0.5, 1.5)$.

prior	estimator	MAE	MSE	bias	estimator MAE	MSE	bias	
0% censoring								
Jeffreys	$\hat{\alpha}_{\text{AE}}$	0.0976	0.0198	0.0356	$\hat{\beta}_{\text{AE}}$	0.6786	0.9781	0.2986
	$\hat{\alpha}_{\text{SE}}$	0.1045	0.0242	0.0501	$\hat{\beta}_{\text{SE}}$	0.795	1.443	0.5097
	$\hat{\alpha}_{\text{LNX(2)}}$	0.0947	0.0178	0.0326	$\hat{\beta}_{\text{LNX(2)}}$	0.4378	0.2957	-0.1002
	$\hat{\alpha}_{\text{GE(2)}}$	0.09	0.0155	0.0092	$\hat{\beta}_{\text{GE(2)}}$	0.5669	0.5369	-0.1029
Reference	$\hat{\alpha}_{\text{AE}}$	0.1004	0.0211	0.038	$\hat{\beta}_{\text{AE}}$	0.7157	1.1389	0.3235
	$\hat{\alpha}_{\text{SE}}$	0.1076	0.0256	0.053	$\hat{\beta}_{\text{SE}}$	0.8327	1.6528	0.541
	$\hat{\alpha}_{\text{LNX(2)}}$	0.0975	0.0192	0.0349	$\hat{\beta}_{\text{LNX(2)}}$	0.4606	0.3311	-0.0918
	$\hat{\alpha}_{\text{GE(2)}}$	0.0926	0.0166	0.0111	$\hat{\beta}_{\text{GE(2)}}$	0.6003	0.6252	-0.0854
PM	$\hat{\alpha}_{\text{AE}}$	0.0909	0.0145	0.0151	$\hat{\beta}_{\text{AE}}$	0.6402	0.8219	0.219
	$\hat{\alpha}_{\text{SE}}$	0.0959	0.0168	0.0287	$\hat{\beta}_{\text{SE}}$	0.7422	1.1708	0.4211
	$\hat{\alpha}_{\text{LNX(2)}}$	0.0892	0.0138	0.0133	$\hat{\beta}_{\text{LNX(2)}}$	0.4427	0.2902	-0.1492
	$\hat{\alpha}_{\text{GE(2)}}$	0.0878	0.0124	-0.0098	$\hat{\beta}_{\text{GE(2)}}$	0.5666	0.4961	-0.1688
20% censoring								
Jeffreys	$\hat{\alpha}_{\text{AE}}$	0.111	0.0176	-0.0731	$\hat{\beta}_{\text{AE}}$	0.7554	1.2113	0.3591
	$\hat{\alpha}_{\text{SE}}$	0.1083	0.0177	-0.0589	$\hat{\beta}_{\text{SE}}$	0.9191	1.8974	0.6366
	$\hat{\alpha}_{\text{LNX(2)}}$	0.1091	0.017	-0.0727	$\hat{\beta}_{\text{LNX(2)}}$	0.4519	0.3091	-0.1254
	$\hat{\alpha}_{\text{GE(2)}}$	0.1221	0.0199	-0.0985	$\hat{\beta}_{\text{GE(2)}}$	0.6024	0.5929	-0.1448
Reference	$\hat{\alpha}_{\text{AE}}$	0.1159	0.0188	-0.07	$\hat{\beta}_{\text{AE}}$	0.7785	1.4654	0.3583
	$\hat{\alpha}_{\text{SE}}$	0.1142	0.0193	-0.0549	$\hat{\beta}_{\text{SE}}$	0.9415	2.283	0.6424
	$\hat{\alpha}_{\text{LNX(2)}}$	0.1136	0.0181	-0.0696	$\hat{\beta}_{\text{LNX(2)}}$	0.4746	0.3458	-0.1403
	$\hat{\alpha}_{\text{GE(2)}}$	0.1248	0.0207	-0.0963	$\hat{\beta}_{\text{GE(2)}}$	0.6451	0.7152	-0.153
PM	$\hat{\alpha}_{\text{AE}}$	0.1212	0.0207	-0.0734	$\hat{\beta}_{\text{AE}}$	0.8086	1.5329	0.3455
	$\hat{\alpha}_{\text{SE}}$	0.1193	0.0212	-0.0587	$\hat{\beta}_{\text{SE}}$	0.963	2.362	0.6207
	$\hat{\alpha}_{\text{LNX(2)}}$	0.1187	0.0199	-0.073	$\hat{\beta}_{\text{LNX(2)}}$	0.4972	0.3832	-0.144
	$\hat{\alpha}_{\text{GE(2)}}$	0.1304	0.0226	-0.0991	$\hat{\beta}_{\text{GE(2)}}$	0.6699	0.7917	-0.1528

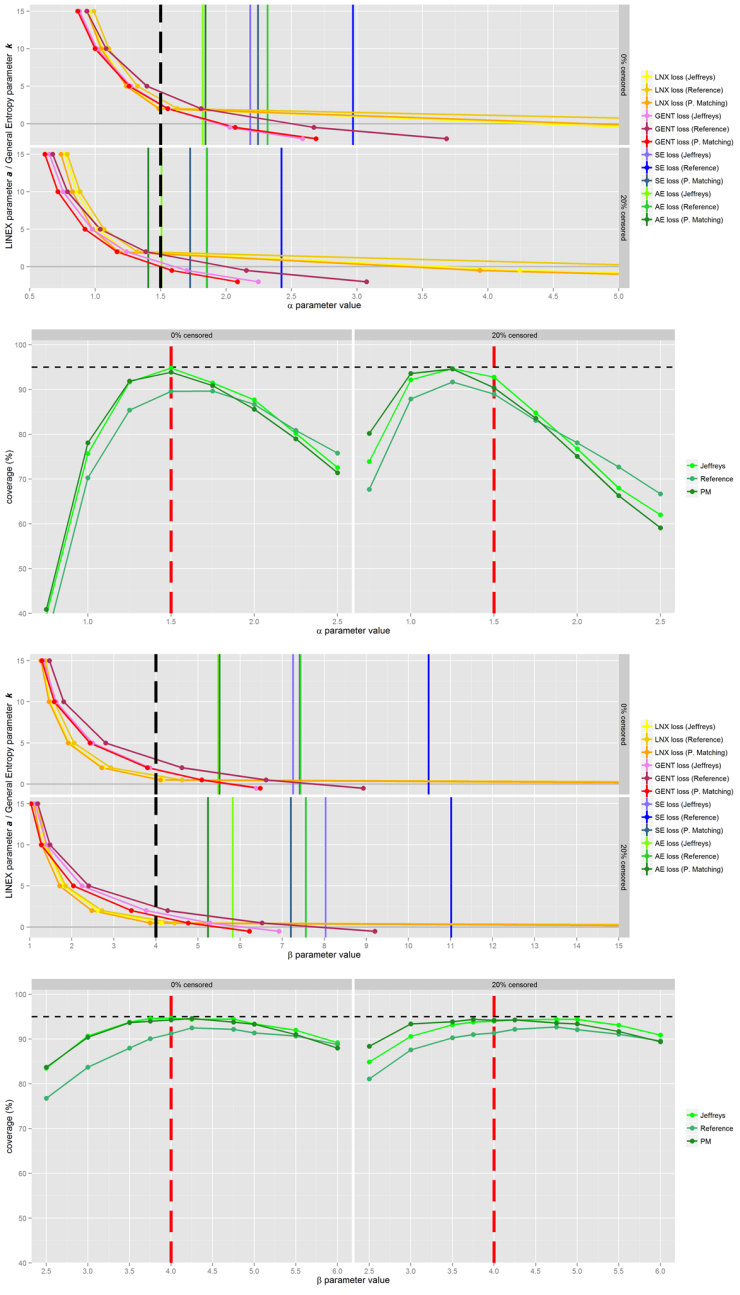


Figure 4.14: *Bayesian estimates and coverages plots for CRG model, with* $\alpha = 1.5$ *(top two) and* $\beta = 4$ *(bottom two), and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours).*

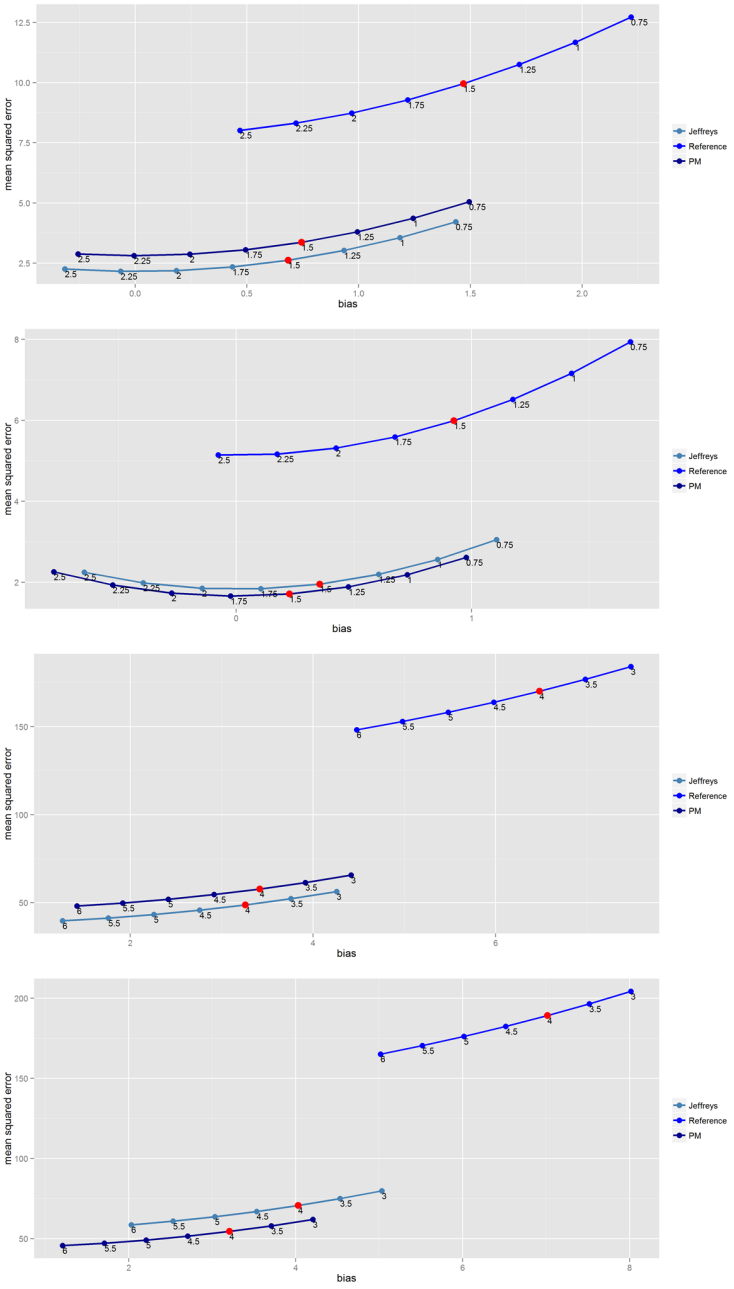


Figure 4.15: *MSE vs bias plots for CRG model, with* $\alpha = 1.5$ *(top two) and* $\beta = 4$ *(bottom two), and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours).*

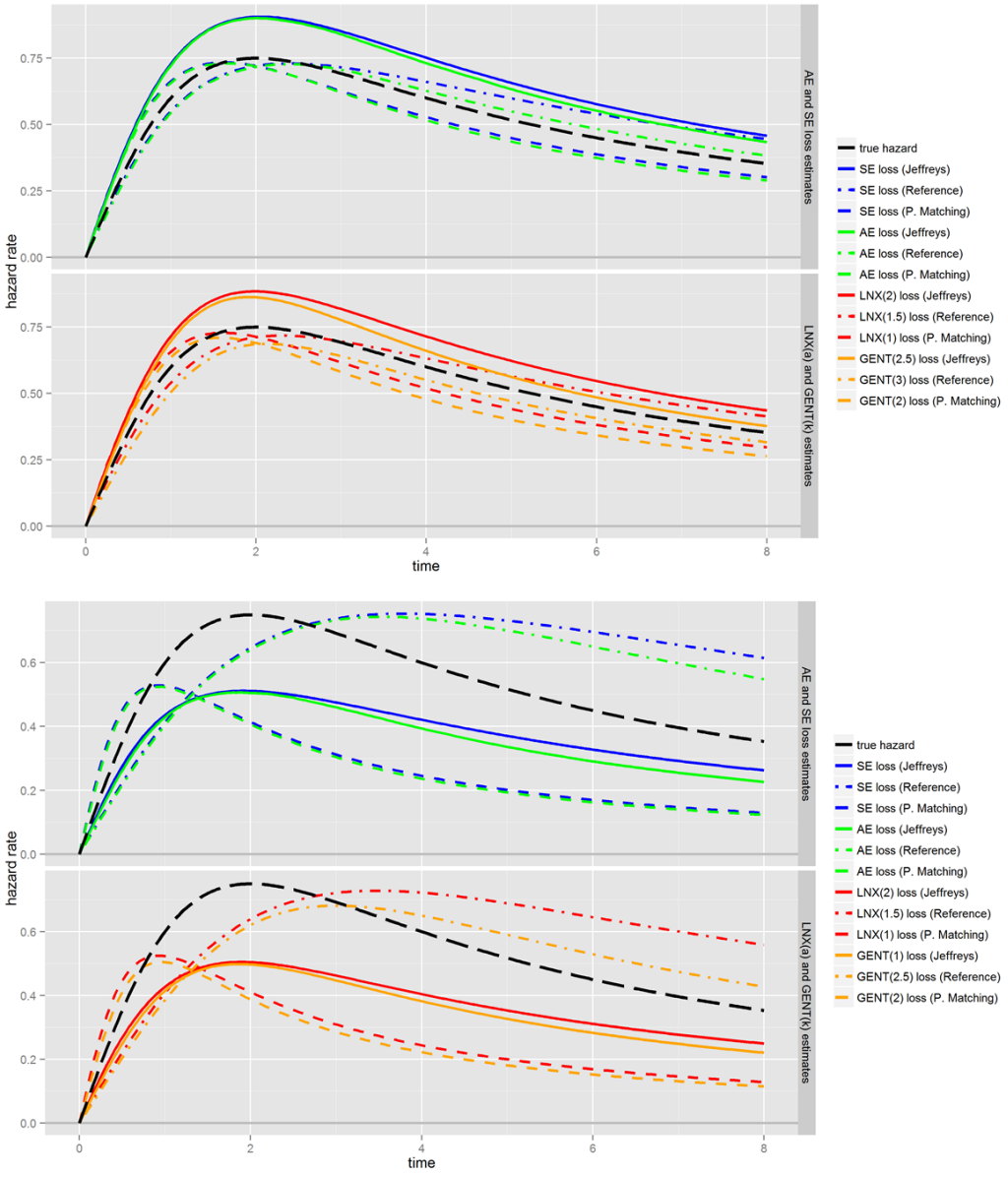


Figure 4.16: *Plots of Bayesian estimates of the hazard function for the CRG model with* $(\alpha, \beta) = (1.5, 4)$ *and two levels of censoring (no censoring, top, and 20% censoring, bottom), derived using three different priors and four different loss functions.*

Table 4.6: *The MAE, MSE and bias of estimators for the CRG model, with parameters* $(\alpha, \beta) = (1.5, 4)$.

prior	estimator	MAE	MSE	bias	estimator MAE	MSE	bias	
0% censoring								
Jeffreys	$\hat{\alpha}_{\text{AE}}$	0.6032	1.0678	0.3224	$\hat{\beta}_{\text{AE}}$	2.548	18.9488	1.4772
	$\hat{\alpha}_{\text{SE}}$	0.8979	2.6234	0.6847	$\hat{\beta}_{\text{SE}}$	3.9983	48.7621	3.2593
	$\hat{\alpha}_{\text{LNX(2)}}$	0.3569	0.2148	-0.0045	$\hat{\beta}_{\text{LNX(2)}}$	1.384	2.4541	-1.2552
	$\hat{\alpha}_{\text{GE(2)}}$	0.4536	0.4356	0.0572	$\hat{\beta}_{\text{GE(2)}}$	1.7352	5.7697	-0.1501
Reference	$\hat{\alpha}_{\text{AE}}$	1.0706	4.8284	0.8175	$\hat{\beta}_{\text{AE}}$	4.3945	81.0112	3.4353
	$\hat{\alpha}_{\text{SE}}$	1.662	9.9552	1.4691	$\hat{\beta}_{\text{SE}}$	7.1406	170.1714	6.48
	$\hat{\alpha}_{\text{LNX(2)}}$	0.4496	0.3941	0.1283	$\hat{\beta}_{\text{LNX(2)}}$	1.3597	2.5153	-1.0759
	$\hat{\alpha}_{\text{GE(2)}}$	0.6581	1.33	0.3063	$\hat{\beta}_{\text{GE(2)}}$	2.3103	16.0758	0.6099
PM	$\hat{\alpha}_{\text{AE}}$	0.644	1.3123	0.3437	$\hat{\beta}_{\text{AE}}$	2.6255	21.4611	1.5148
	$\hat{\alpha}_{\text{SE}}$	0.9773	3.3653	0.7442	$\hat{\beta}_{\text{SE}}$	4.1958	57.8186	3.4172
	$\hat{\alpha}_{\text{LNX(2)}}$	0.3663	0.2218	-0.0155	$\hat{\beta}_{\text{LNX(2)}}$	1.4107	2.5176	-1.2904
	$\hat{\alpha}_{\text{GE(2)}}$	0.4695	0.4618	0.0517	$\hat{\beta}_{\text{GE(2)}}$	1.7353	5.5651	-0.2051
20% censoring								
Jeffreys	$\hat{\alpha}_{\text{AE}}$	0.6002	0.8071	0.0059	$\hat{\beta}_{\text{AE}}$	2.9661	26.902	1.828
	$\hat{\alpha}_{\text{SE}}$	0.8193	1.9544	0.3552	$\hat{\beta}_{\text{SE}}$	4.7636	70.7126	4.0316
	$\hat{\alpha}_{\text{LNX(2)}}$	0.4346	0.2551	-0.2673	$\hat{\beta}_{\text{LNX(2)}}$	1.5008	2.8094	-1.3782
	$\hat{\alpha}_{\text{GE(2)}}$	0.5171	0.3899	-0.263	$\hat{\beta}_{\text{GE(2)}}$	1.8995	6.6705	-0.2383
Reference	$\hat{\alpha}_{\text{AE}}$	0.903	2.8386	0.3521	$\hat{\beta}_{\text{AE}}$	4.6856	87.1115	3.5622
	$\hat{\alpha}_{\text{SE}}$	1.3516	5.99	0.9239	$\hat{\beta}_{\text{SE}}$	7.7649	189.2172	7.0173
	$\hat{\alpha}_{\text{LNX(2)}}$	0.4691	0.3231	-0.1796	$\hat{\beta}_{\text{LNX(2)}}$	1.4907	2.8625	-1.2883
	$\hat{\alpha}_{\text{GE(2)}}$	0.6065	0.7102	-0.1149	$\hat{\beta}_{\text{GE(2)}}$	2.3295	13.5155	0.2743
PM	$\hat{\alpha}_{\text{AE}}$	0.5727	0.7242	-0.094	$\hat{\beta}_{\text{AE}}$	2.5482	20.3995	1.2397
	$\hat{\alpha}_{\text{SE}}$	0.744	1.7082	0.2261	$\hat{\beta}_{\text{SE}}$	4.0385	54.4698	3.2072
	$\hat{\alpha}_{\text{LNX(2)}}$	0.4567	0.2733	-0.325	$\hat{\beta}_{\text{LNX(2)}}$	1.6112	3.1318	-1.525
	$\hat{\alpha}_{\text{GE(2)}}$	0.5347	0.3972	-0.3349	$\hat{\beta}_{\text{GE(2)}}$	1.8451	5.8301	-0.5862

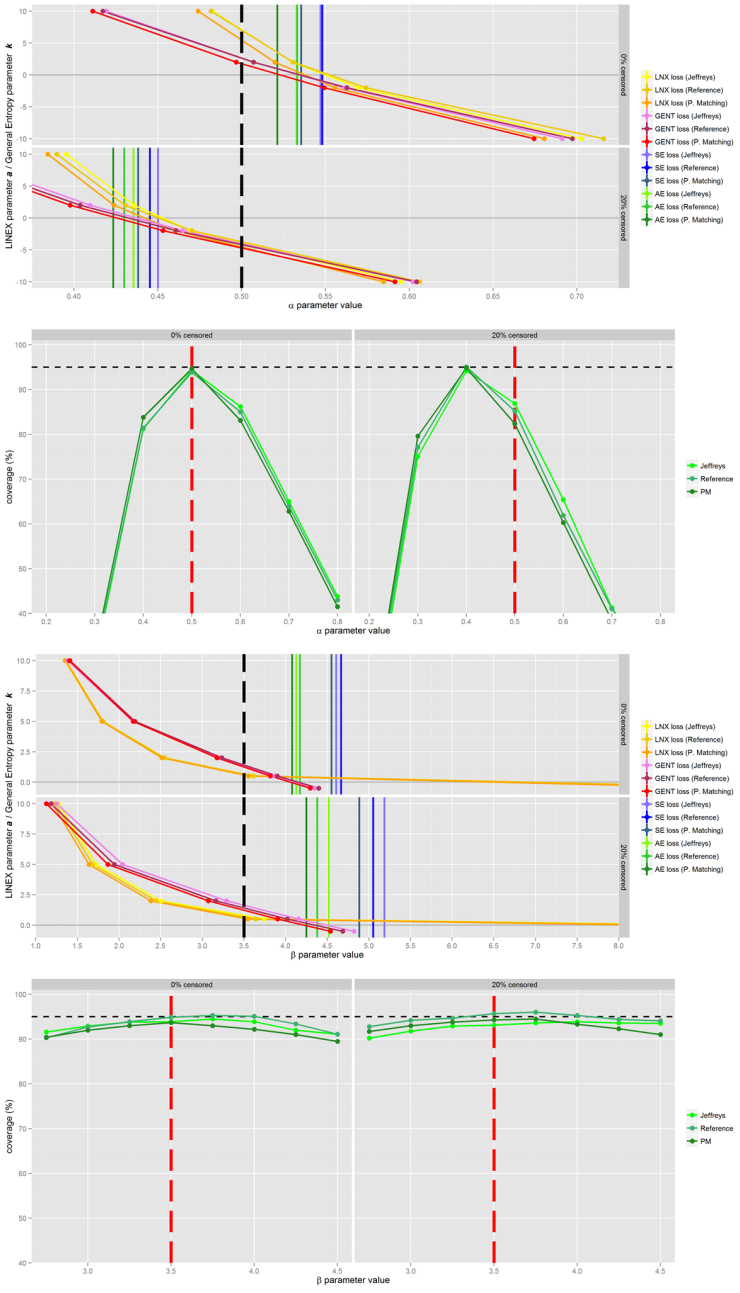


Figure 4.17: *Bayesian estimates and coverages plots for CRG model, with* $\alpha = 0.5$ *(top two) and* $\beta = 3.5$ *(bottom two), and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours).*

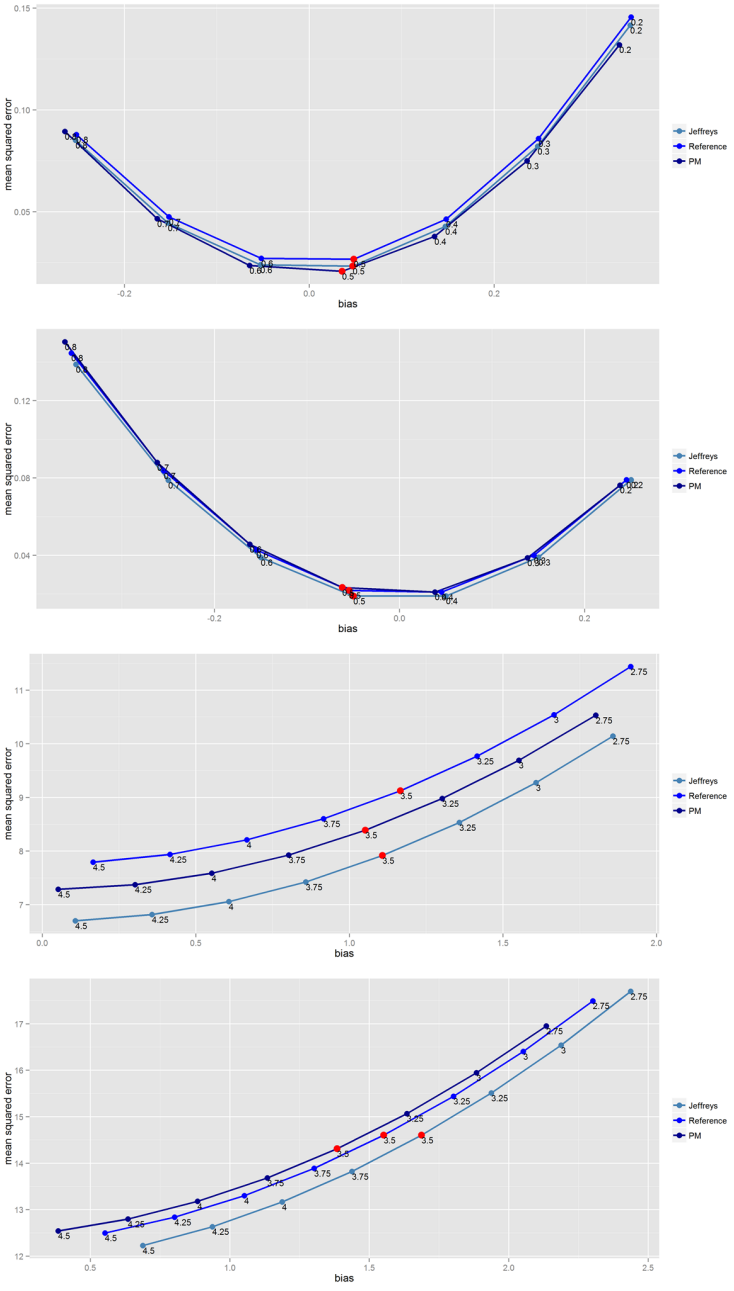


Figure 4.18: *MSE vs bias plots for CRG model, with* $\alpha = 0.5$ *(top two) and* $\beta = 3.5$ *(bottom two), and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours).*

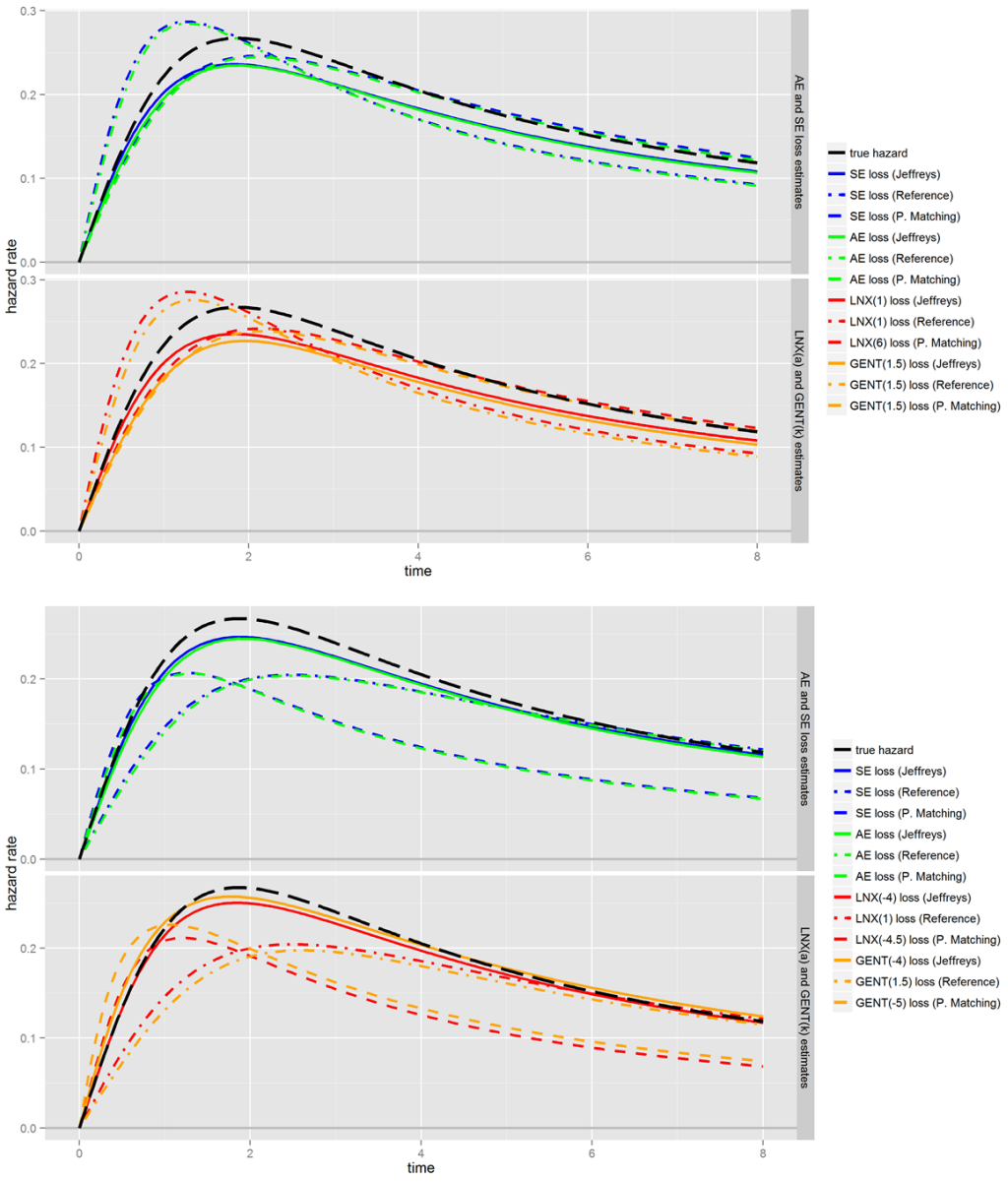


Figure 4.19: *Plots of Bayesian estimates of the hazard function for the CRG model with* $(\alpha, \beta) = (0.5, 3.5)$ *and two levels of censoring (no censoring, top, and 20% censoring, bottom), derived using three different priors and four different loss functions.*

Table 4.7: *The MAE, MSE and bias of estimators for the CRG model, with parameters* $(\alpha, \beta) = (0.5, 3.5)$.

prior	estimator	MAE	MSE	bias	estimator MAE	MSE	bias	
0% censoring								
Jeffreys	$\hat{\alpha}_{\text{AE}}$	0.0969	0.0196	0.0329	$\hat{\beta}_{\text{AE}}$	1.6002	5.6083	0.629
	$\hat{\alpha}_{\text{SE}}$	0.1032	0.0233	0.0471	$\hat{\beta}_{\text{SE}}$	1.8438	7.9171	1.1077
	$\hat{\alpha}_{\text{LNX(2)}}$	0.0941	0.0179	0.0299	$\hat{\beta}_{\text{LNX(2)}}$	1.1377	1.7264	-0.9708
	$\hat{\alpha}_{\text{GE(2)}}$	0.0903	0.0156	0.0066	$\hat{\beta}_{\text{GE(2)}}$	1.3745	3.2072	-0.3016
Reference	$\hat{\alpha}_{\text{AE}}$	0.1007	0.0219	0.0332	$\hat{\beta}_{\text{AE}}$	1.6309	6.3735	0.6731
	$\hat{\alpha}_{\text{SE}}$	0.1073	0.0267	0.0481	$\hat{\beta}_{\text{SE}}$	1.8841	9.1247	1.1659
	$\hat{\alpha}_{\text{LNX(2)}}$	0.0977	0.0197	0.0302	$\hat{\beta}_{\text{LNX(2)}}$	1.1521	1.7638	-0.9557
	$\hat{\alpha}_{\text{GE(2)}}$	0.0944	0.0174	0.0066	$\hat{\beta}_{\text{GE(2)}}$	1.3939	3.6344	-0.27
PM	$\hat{\alpha}_{\text{AE}}$	0.0931	0.0175	0.0213	$\hat{\beta}_{\text{AE}}$	1.6221	6.0032	0.5795
	$\hat{\alpha}_{\text{SE}}$	0.0984	0.0207	0.0355	$\hat{\beta}_{\text{SE}}$	1.8421	8.3913	1.052
	$\hat{\alpha}_{\text{LNX(2)}}$	0.0909	0.0163	0.0193	$\hat{\beta}_{\text{LNX(2)}}$	1.1961	1.8539	-0.9877
	$\hat{\alpha}_{\text{GE(2)}}$	0.089	0.0146	-0.0037	$\hat{\beta}_{\text{GE(2)}}$	1.4403	3.599	-0.3283
20% censoring								
Jeffreys	$\hat{\alpha}_{\text{AE}}$	0.1111	0.0182	-0.0645	$\hat{\beta}_{\text{AE}}$	1.8923	9.543	1.0173
	$\hat{\alpha}_{\text{SE}}$	0.1089	0.0189	-0.0498	$\hat{\beta}_{\text{SE}}$	2.3013	14.6037	1.6874
	$\hat{\alpha}_{\text{LNX(2)}}$	0.109	0.0174	-0.0645	$\hat{\beta}_{\text{LNX(2)}}$	1.206	1.8701	-0.9971
	$\hat{\alpha}_{\text{GE(2)}}$	0.121	0.0198	-0.0908	$\hat{\beta}_{\text{GE(2)}}$	1.5255	4.5872	-0.2133
Reference	$\hat{\alpha}_{\text{AE}}$	0.1154	0.0204	-0.07	$\hat{\beta}_{\text{AE}}$	1.7673	9.0697	0.8802
	$\hat{\alpha}_{\text{SE}}$	0.1136	0.0219	-0.0547	$\hat{\beta}_{\text{SE}}$	2.1727	14.6039	1.5519
	$\hat{\alpha}_{\text{LNX(2)}}$	0.1128	0.019	-0.07	$\hat{\beta}_{\text{LNX(2)}}$	1.2137	1.8972	-1.0572
	$\hat{\alpha}_{\text{GE(2)}}$	0.1254	0.0217	-0.0967	$\hat{\beta}_{\text{GE(2)}}$	1.4586	4.2842	-0.3398
PM	$\hat{\alpha}_{\text{AE}}$	0.1197	0.0223	-0.0765	$\hat{\beta}_{\text{AE}}$	1.8348	9.6103	0.7511
	$\hat{\alpha}_{\text{SE}}$	0.1166	0.0234	-0.0617	$\hat{\beta}_{\text{SE}}$	2.1828	14.314	1.3844
	$\hat{\alpha}_{\text{LNX(2)}}$	0.1166	0.0203	-0.0765	$\hat{\beta}_{\text{LNX(2)}}$	1.2585	2.0571	-1.1196
	$\hat{\alpha}_{\text{GE(2)}}$	0.1302	0.0232	-0.1027	$\hat{\beta}_{\text{GE(2)}}$	1.533	4.4612	-0.4307

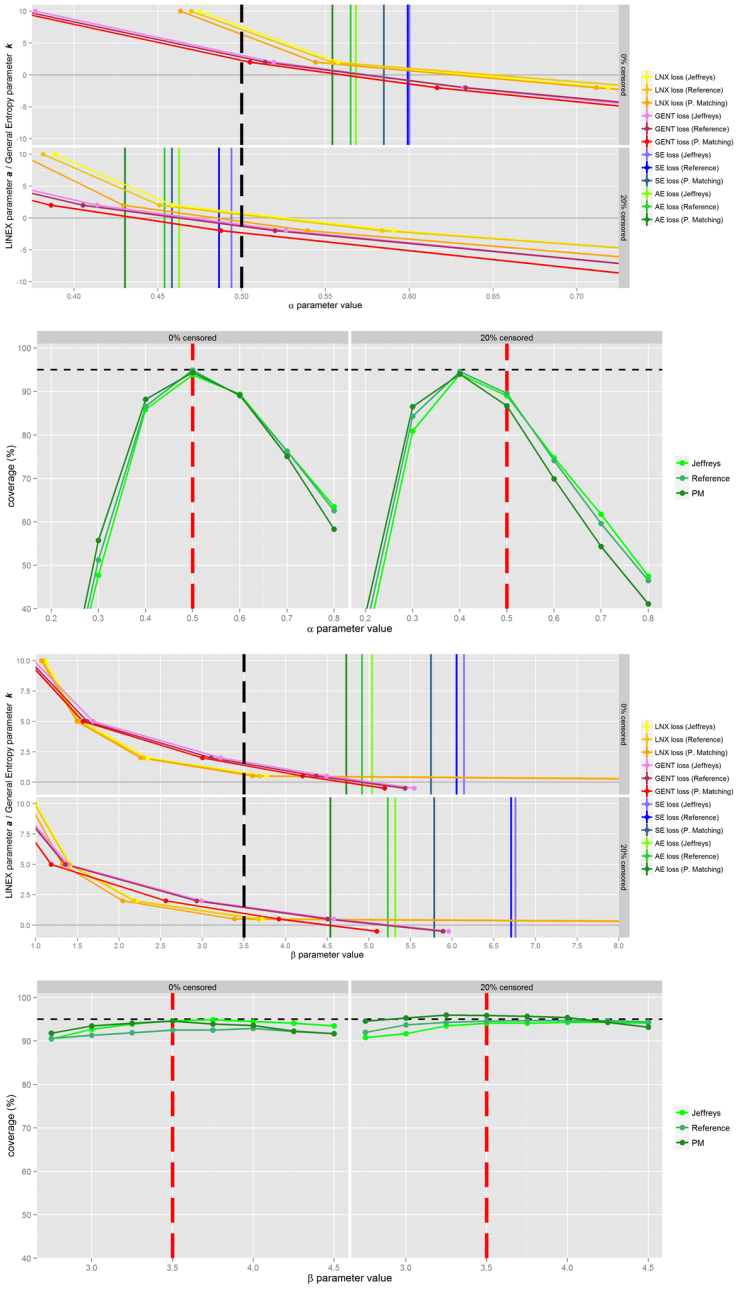


Figure 4.20: *Bayesian estimates and coverages plots for CRG model, with* $\alpha = 0.5$ *(top two) and* $\beta = 3.5$ *(bottom two), and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours), and with sample size* $n = 30$.

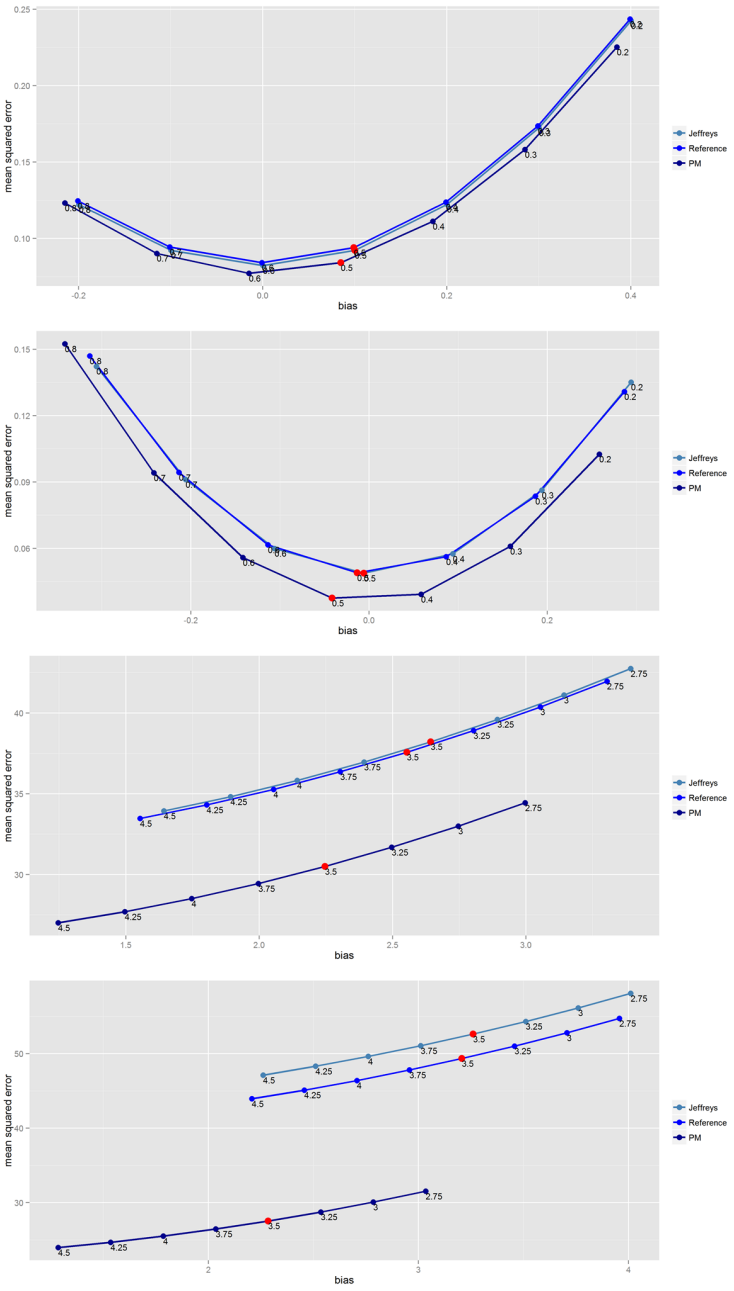


Figure 4.21: *MSE vs bias plots for CRG model, with* $\alpha = 0.5$ *(top two) and* $\beta = 3.5$ *(bottom two), and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours), and with sample size* $n = 30$.

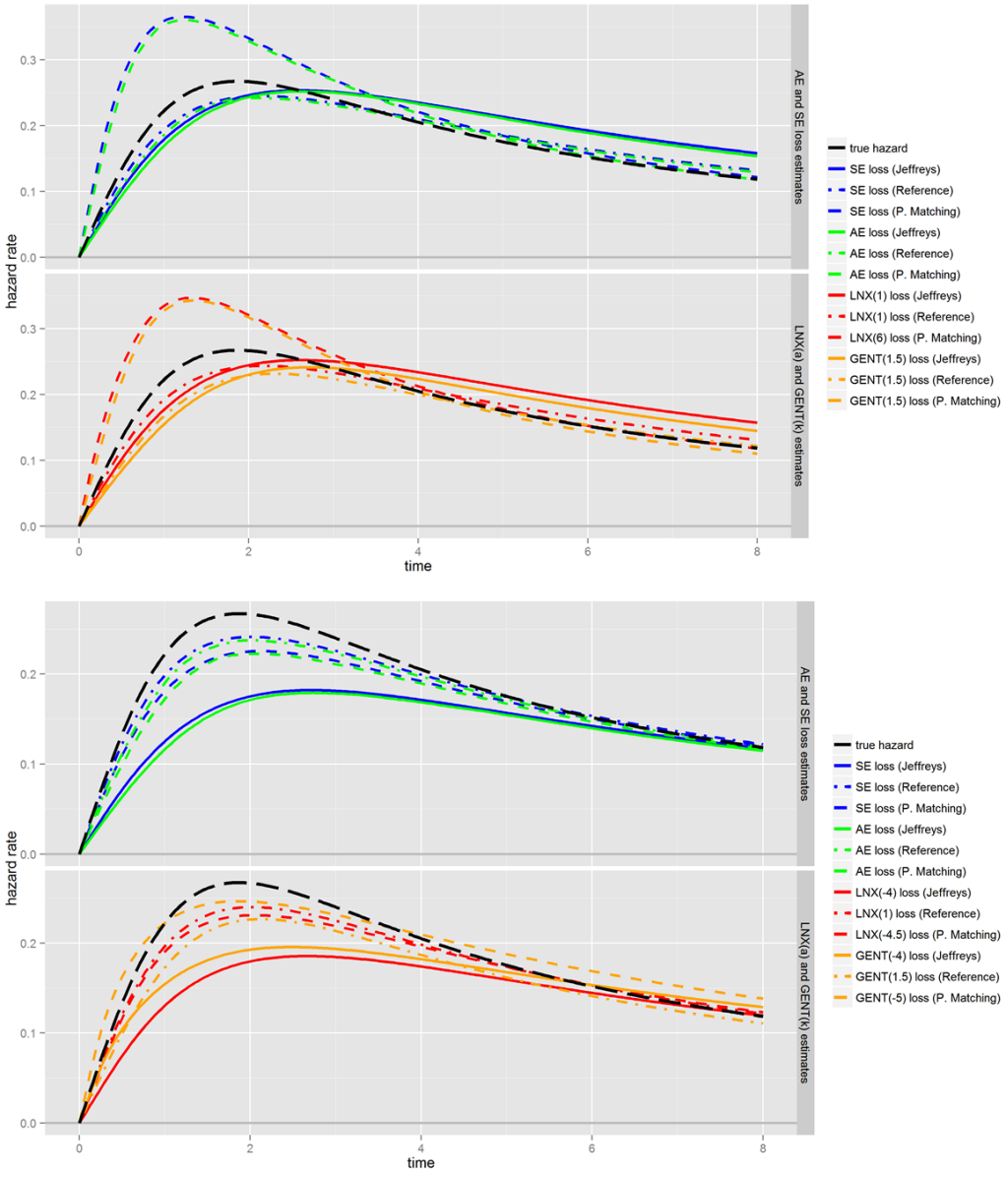


Figure 4.22: *Plots of Bayesian estimates of the hazard function for the CRG model with* $(\alpha, \beta) = (0.5, 3.5)$ *and two levels of censoring (no censoring, top, and 20% censoring, bottom), derived using three different priors, four different loss functions and sample size* $n = 30$.

Table 4.8: *The MAE, MSE and bias of estimators for the CRG model, with parameters* $(\alpha, \beta) = (0.5, 3.5)$ *and* $n = 30$.

prior	estimator	MAE	MSE	bias	estimator MAE	MSE	bias	
0% censoring								
Jeffreys	$\hat{\alpha}_{\text{AE}}$	0.1425	0.0619	0.0682	$\hat{\beta}_{\text{AE}}$	2.5441	22.167	1.54
	$\hat{\alpha}_{\text{SE}}$	0.1639	0.0923	0.1002	$\hat{\beta}_{\text{SE}}$	3.3341	38.218	2.6426
	$\hat{\alpha}_{\text{LNX(2)}}$	0.1312	0.0397	0.0571	$\hat{\beta}_{\text{LNX(2)}}$	1.3277	2.2991	-1.1554
	$\hat{\alpha}_{\text{GE(2)}}$	0.1217	0.0364	0.0186	$\hat{\beta}_{\text{GE(2)}}$	1.8334	7.49	-0.2821
Reference	$\hat{\alpha}_{\text{AE}}$	0.1438	0.0619	0.0651	$\hat{\beta}_{\text{AE}}$	2.476	21.1967	1.4181
	$\hat{\alpha}_{\text{SE}}$	0.1656	0.0941	0.0991	$\hat{\beta}_{\text{SE}}$	3.2884	37.5738	2.5536
	$\hat{\alpha}_{\text{LNX(2)}}$	0.1315	0.0388	0.0532	$\hat{\beta}_{\text{LNX(2)}}$	1.3684	2.3629	-1.2065
	$\hat{\alpha}_{\text{GE(2)}}$	0.1235	0.0348	0.0135	$\hat{\beta}_{\text{GE(2)}}$	1.793	6.8769	-0.3992
PM	$\hat{\alpha}_{\text{AE}}$	0.0931	0.0175	0.0213	$\hat{\beta}_{\text{AE}}$	1.6221	6.0032	0.5795
	$\hat{\alpha}_{\text{SE}}$	0.0984	0.0207	0.0355	$\hat{\beta}_{\text{SE}}$	1.8421	8.3913	1.052
	$\hat{\alpha}_{\text{LNX(2)}}$	0.0909	0.0163	0.0193	$\hat{\beta}_{\text{LNX(2)}}$	1.1961	1.8539	-0.9877
	$\hat{\alpha}_{\text{GE(2)}}$	0.089	0.0146	-0.0037	$\hat{\beta}_{\text{GE(2)}}$	1.4403	3.599	-0.3283
20% censoring								
Jeffreys	$\hat{\alpha}_{\text{AE}}$	0.1421	0.0365	-0.0372	$\hat{\beta}_{\text{AE}}$	2.8966	28.9976	1.8168
	$\hat{\alpha}_{\text{SE}}$	0.1501	0.0487	-0.006	$\hat{\beta}_{\text{SE}}$	3.9471	52.6457	3.2606
	$\hat{\alpha}_{\text{LNX(2)}}$	0.136	0.0309	-0.0403	$\hat{\beta}_{\text{LNX(2)}}$	1.4256	2.5816	-1.2788
	$\hat{\alpha}_{\text{GE(2)}}$	0.1457	0.0319	-0.0857	$\hat{\beta}_{\text{GE(2)}}$	1.9882	8.9625	-0.4382
Reference	$\hat{\alpha}_{\text{AE}}$	0.1415	0.0367	-0.046	$\hat{\beta}_{\text{AE}}$	2.8239	27.0935	1.7279
	$\hat{\alpha}_{\text{SE}}$	0.1488	0.0489	-0.0134	$\hat{\beta}_{\text{SE}}$	3.9041	49.3627	3.2068
	$\hat{\alpha}_{\text{LNX(2)}}$	0.1335	0.029	-0.0494	$\hat{\beta}_{\text{LNX(2)}}$	1.4495	2.6165	-1.323
	$\hat{\alpha}_{\text{GE(2)}}$	0.1468	0.0312	-0.0951	$\hat{\beta}_{\text{GE(2)}}$	1.9303	7.4233	-0.5704
PM	$\hat{\alpha}_{\text{AE}}$	0.1407	0.0306	-0.0696	$\hat{\beta}_{\text{AE}}$	2.3082	13.7386	1.039
	$\hat{\alpha}_{\text{SE}}$	0.1433	0.0375	-0.0416	$\hat{\beta}_{\text{SE}}$	3.114	27.5445	2.2848
	$\hat{\alpha}_{\text{LNX(2)}}$	0.1346	0.0268	-0.071	$\hat{\beta}_{\text{LNX(2)}}$	1.5249	2.8245	-1.4587
	$\hat{\alpha}_{\text{GE(2)}}$	0.1521	0.0306	-0.1141	$\hat{\beta}_{\text{GE(2)}}$	1.8737	5.0001	-0.9394

4.2.5 Discussion

The discussion of the results in the previous section now follows and the effect of censoring is considered first. Two levels of censoring were used, 0% and 20%, and while an increase in the variance of the estimators are observed, it is not as prominent as with the CRE model in Section 4.1. In this case, it is clear that censoring causes underestimation of the α parameter in addition to a slight overestimation of β. However, there does not seem to be much of a difference apart from a slight increase (in some cases) in the MSE and bias.

The nature of the estimation errors become clearer in the coverage values and MSE-bias plots. In general, the estimates for parameter α with the best frequentist properties (highest coverage) in the censored cases are about 80% of the true value. On the other hand, there seems to be an inherent overestimation of β that occurs even with no censoring; this is only increased slightly with censoring.

Different Bayesian estimators were also used and those derived with a symmetric loss function are considered first. For the α parameter, with no censoring, $\hat{\alpha}_{\text{AE}}$ is more accurate than $\hat{\alpha}_{\text{SE}}$, as was seen with the CRE model parameter. However, an interesting thing happens with censoring. Since all estimations are now skewed negatively, $\hat{\alpha}_{\text{SE}}$ is now closer to the true values. For β, it is clear that $\hat{\beta}_{\text{AE}}$ is more accurate than $\hat{\beta}_{\text{SE}}$, as overestimation is more severe in the latter. These observations are all emphasised with the MAE, MSE and bias calculated across all parameter configurations.

The estimators derived using the two asymmetric loss functions are also similar to what was seen in Section 4.1.4. Both LINEX and GE estimators produce a hyperbolic-shaped monotonically decreasing curve for varying values of their loss parameters, although the latter to a lesser extent. The value of k which optimises the accuracy of $\hat{\alpha}_{\text{GE(k)}}$ stays more stable for different α configurations than is the case with $\hat{\alpha}_{\text{LNX(a)}}$. In general, it seems that $\hat{\beta}_{\text{GE(k)}}$ and $\hat{\beta}_{\text{LNX(a)}}$ is very similar in this regard.

The performance of the estimators are compared with regards to different priors used in their derivation. From all results, it seems that the Jeffreys prior leads to the most accurate and best performance, closely followed by the PM prior. This is the case for both model parameters. Even though the PM prior sometimes leads to lower MAE, MSE and bias of the estimators in the non-censored case, the Jeffreys prior usually obtains the

best performance in the presence of censoring. It appears that the reference prior leads to inferior performance, especially with the parameter configuration of $(\alpha, \beta) = (1.5, 4)$, where all accuracy measures are much higher than those relating to the other priors. For other parameter configurations, the reference prior sometimes leads to lower MAE, MSE and bias measures than the PM prior.

In general, the estimators of the hazard rate reproduced the general shape of the true hazard rate fairly well in the non-censored case and there seemed to be little discrepancy between the symmetric loss functions, as well as between the asymmetric loss functions. The prior distribution played a much larger role in determining the overall shape of the estimated hazard. Different priors yielded the closest estimates for different parameter configurations, but overall those derived with the Jeffreys prior appeared to be the most stable for both non-censored and censored scenarios.

Finally, the effect of varying sample size was also investigated. As expected, it would seem that an increase in sample size improves the accuracy and precision of the estimators, but the trends and general conclusions discussed above remains unchanged.

In the Chapter 5, the compound Rayleigh models investigated in this thesis are subjected to a generalisation in order to increase flexibility of their forms, and a similar simulation procedure is used to assess their performance.

4.3 Summary

This chapter explores the CRE and CRG models and their respective simulation studies. The derivations regarding the characteristics of these models will be summarised in this section.

	CRE model	CRG model
survival function	$S(t,\gamma) = \left(1 + \frac{t^2}{\gamma}\right)^{-1}$	$S(t,\alpha,\beta) = \beta^{\alpha}(t^2+\beta)^{-\alpha}$
hazard rate	$h(t,\gamma) = \frac{2t}{t^2+\gamma}$	$h(t,\alpha,\beta) = \frac{2\alpha t}{t^2+\beta}$
Fisher information	$\frac{1}{3\gamma^2}$	$\begin{bmatrix} \frac{1}{\alpha^2} & \frac{-1}{\beta(\alpha+1)} \\ \frac{-1}{\beta(\alpha+1)} & \frac{\alpha}{\beta^2(\alpha+2)} \end{bmatrix}$
likelihood	$\mathcal{L}(\gamma\|\mathbf{t}) \propto \gamma^n e^{W_1(\gamma)-W_2(\gamma)}$ where $W_1(\gamma) = \sum_{i=1}^{d} \ln\left(\frac{t_i}{t_i^2+\gamma}\right)$ and $W_2(\gamma) = \sum_{j=1}^{n} \ln(t_j^2+\gamma)$	$\mathcal{L}(\alpha,\beta\|\mathbf{t}) \propto (2\alpha)^d e^{W_1(\beta)-\alpha W_2(\beta)}$ where $W_1(\beta) = \sum_{i=1}^{d} \ln\left(\frac{t_i}{t_i^2+\beta}\right)$ and $W_2(\beta) = \sum_{i=1}^{n} \ln\left(1+\frac{t_i^2}{\beta}\right)$

Take note that the likelihood functions are defined for a given sample of n survival times $\mathbf{t} = (t_1, t_2, \ldots, t_n)$ ordered such that the first d are non-censored and the remaining $(n-d)$ right censored.

Lastly, the non-informative prior distributions for the two models are tabled. Recall that for the CRE model, only the Jeffreys prior is derived, since for single-parameter models the reference and PM priors are equivalent.

	CRE model	CRG model
Jeffreys prior	$\pi_{\text{jeff}}(\gamma) \propto \frac{1}{\gamma}$	$\pi_{\text{jeff}}(\alpha, \beta) \propto \frac{1}{\beta(\alpha+1)\sqrt{\alpha(\alpha+2)}}$
reference prior		$\pi(\alpha, \beta)_{\text{ref}} \propto \frac{1}{\alpha\beta}$
PM prior		$\pi_{\text{PM}}(\alpha, \beta) \propto \frac{1}{\alpha\beta(\alpha+1)}$

The exposition of results for the simulation study and corresponding discussion can be found in Sections 4.1.4 and 4.1.5 for the CRE model, and Sections 4.2.4 and 4.2.5 for the CRG model.

Chapter 5

Simulation study of compounded and generalised models

The previous chapter dealt with the simulation study of the CRE and CRG models. This chapter continuous in similar vein, but for the generalised counterparts of these models, i.e. the GCRE and GCRG models. Their survival and hazard functions, likelihood functions, Fisher information matrices and Bayesian estimators are derived, as well as the relevant non-informative prior distributions. Thereafter, the simulation study results are shown and discussed.

5.1 The GCRE model

5.1.1 Model characteristics

The versatility of the compound Rayleigh models that were investigated in the previous chapter is extended through a generalisation. This procedure is discussed in Section 3.2.2.3 where specific generalised models are derived. The GCRE distribution, with PDF (3.11) and CDF (3.9), is the focus of this section.

Firstly, the survival function and hazard rate can be derived from (3.9), resulting in

$$S(t, \gamma, c) = \left(1 + \frac{t^c}{\gamma}\right)^{-1}$$

and

$$h(t,\gamma,c) = \frac{ct^{c-1}}{t^c+\gamma}. \tag{5.1}$$

It is clear that the forms of these functions are similar to the non-generalised case in Section 4.1, albeit with an extra parameter controlling the exponentiation of the observations. The additional flexibility of the hazard rate attained through the generalisation is portrayed in Figure 5.1.

Secondly, the Fisher information matrix (2.11) needs to be constructed. This requires the second-order partial derivatives of l_f with respect to γ and c, where

$$l_f = \ln f(t|\gamma,c) = \ln c + \ln\gamma + (c-1)\ln t - 2\ln(t^c+\gamma).$$

It can easily be shown that

$$\begin{aligned}
\frac{\partial^2 l_f}{\partial\gamma^2} &= \frac{2}{(t^c+\gamma)^2} - \frac{1}{\gamma^2} \\
\frac{\partial^2 l_f}{\partial\gamma\partial c} &= \frac{2t^c\ln t}{(t^c+\gamma)^2} \\
\frac{\partial^2 l_f}{\partial c^2} &= -\frac{1}{c^2} - \frac{2\gamma t^c(\ln t)^2}{(t^c+\gamma)^2}.
\end{aligned}$$

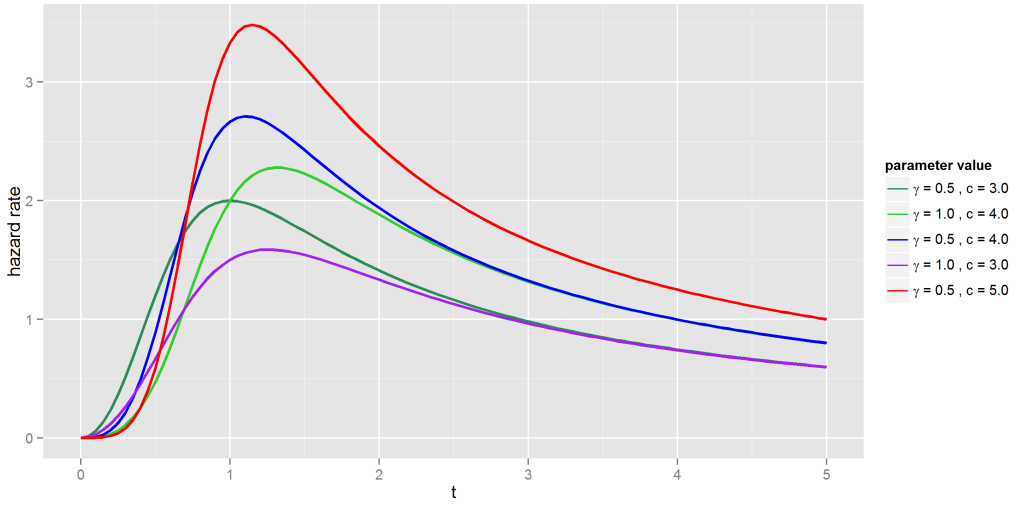


Figure 5.1: *Hazard rate of GCRE model for various values of its parameters.*

Thus, the Fisher information becomes

$$\mathcal{I}_F = \mathrm{E}_{T|\gamma,c}\begin{bmatrix} \frac{1}{\gamma^2} - \frac{2}{(t^c+\gamma)^2} & -\frac{2t^c \ln t}{(t^c+\gamma)^2} \\ -\frac{2t^c \ln t}{(\gamma+t^c)^2} & \frac{1}{c^2} + \frac{2\gamma t^c (\ln t)^2}{(t^c+\gamma)^2} \end{bmatrix}. \tag{5.2}$$

The expected value only needs to be evaluated over terms that contain the survival time, i.e. $\mathrm{E}\left[\frac{1}{(T^c+\gamma)^2}\right]$, $\mathrm{E}\left[\frac{T^c \ln T}{(T^c+\gamma)^2}\right]$ and $\mathrm{E}\left[\frac{T^c (\ln T)^2}{(T^c+\gamma)^2}\right]$. The solution of the first expectation uses the PDF form of the Beta Prime distribution for simplification (refer to Appendix A.2). Accordingly, using (3.11), the expectation is rewritten as

$$\begin{aligned} \mathrm{E}\left[\frac{1}{(T^c+\gamma)^2}\right] &= \int_0^\infty c\gamma t^{c-1}(t^c+\gamma)^{-4} dt \\ &= \int_0^\infty c\gamma^{-3}\left(\gamma^{\frac{1}{c}}\right)^{c-1}\left(\gamma^{\frac{1}{c}}\right)^{1-c} t^{c-1}\left(1+\frac{t^c}{\gamma}\right)^{-4} dt \\ &= \gamma^{-2-\frac{1}{c}} \int_0^\infty c\left(\frac{t}{\gamma^{\frac{1}{c}}}\right)^{c-1}\left[1+\left(\frac{t}{\gamma^{\frac{1}{c}}}\right)^c\right]^{-4} dt \\ &= \gamma^{-2-\frac{1}{c}}\gamma^{\frac{1}{c}} B(1,3), \end{aligned}$$

where, in the third step, a Beta Prime distribution with parameters $p \equiv c$, $q \equiv \gamma^{\frac{1}{c}}$, $r \equiv 1$, and $s \equiv 3$ were used. In consequence, from the definition of the Beta function it follows that

$$\mathrm{E}\left[\frac{1}{(T^c+\gamma)^2}\right] = \frac{1}{3\gamma^2}.$$

The remaining two expectations do not have a similar analytically tractable solution. Instead, consider the simplifications

$$\begin{aligned} \mathrm{E}\left[\frac{T^c \ln T}{(T^c+\gamma)^2}\right] &= \int_0^\infty c\gamma \frac{t^{2c-1}\ln t}{(t^c+\gamma)^4} dt &&= A_1(\gamma, c) \\ \mathrm{E}\left[\frac{T^c (\ln T)^2}{(T^c+\gamma)^2}\right] &= \int_0^\infty c\gamma \frac{t^{2c-1}(\ln t)^2}{(t^c+\gamma)^4} dt &&= A_2(\gamma, c) \end{aligned}$$

where the insoluble integrals (for the moment) are denoted by $A_1(\gamma, c)$ and $A_2(\gamma, c)$. With these assignments, the form of the Fisher information matrix (5.2) can be stated as

$$\mathcal{I}_F = \begin{bmatrix} \frac{1}{3\gamma^2} & -2A_1(\gamma, c) \\ -2A_1(\gamma, c) & \frac{1}{c^2} + 2A_2(\gamma, c) \end{bmatrix}. \tag{5.3}$$

In order to follow through with the derivation of non-informative prior distributions, numerical approximation as described in Section 2.2.3.2 will be used. More specifically, the porportional form of the posterior distribution, which is derived using $\mathcal{I}_F$, is required in the calculation of the MCMC algorithm. The solution formulated above is not closed-form, but dependent on the terms $A_1(\gamma, c)$ and $A_2(\gamma, c)$. These need to be evaluated thousands of times during the course of the simulation run and each time, the adaptive quadrature numerical integration routine is called for approximation with a high degree of accuracy. With this kept in mind, the posterior distribution can be derived in the next section using the simplified form of $\mathcal{I}_F$ above.

5.1.2 Prior and posterior distributions

A posterior for the GCRE model can be found by specification of a prior and derivation of the likelihood. Even though this model has two parameters, only the Jeffreys prior is taken into consideration here. This is due to reasons such as brevity and computational complexity of the numerical approximations used in the calculation of the Fisher information matrix. This model can also be regarded as a special case of the GCRG model, for which priors are explored in more detail in Section 5.2.2.

Jeffrey's prior distribution can be derived as the square root of the determinant. From Section 2.2.4.1,

$$\pi_{\text{jeff}}(\gamma, c) \propto \sqrt{|\mathcal{I}_F|}$$

and using (5.3), the prior becomes

$$\pi_{\text{jeff}}(\gamma, c) \propto \sqrt{\frac{1}{3\gamma^2}\left(\frac{1}{c^2} + 2A_2(\gamma, c)\right) - 4\gamma c\left(A_1(\gamma, c)\right)^2}. \tag{5.4}$$

Next, the likelihood function can be derived by considering a sample of n survival times $\mathbf{t} = (t_1, t_2, \ldots, t_n)$, ordered such that the first d are non-censored and the remaining

$(n-d)$ right censored. Then

$$\begin{aligned}\mathcal{L}(\gamma, c|\mathbf{t}) &\propto \prod_{i=1}^{d} f(t_i|\gamma, c) \prod_{j=d+1}^{n} S(t_j, \gamma, c) \\ &\propto \prod_{i=1}^{d} \frac{c\gamma t_i^{c-1}}{(t_i^c+\gamma)^2} \prod_{j=d+1}^{n} \frac{\gamma}{(t_j^c+\gamma)} \\ &\propto c^d \gamma^n \prod_{i=1}^{d} \frac{t_i^{c-1}}{(t_i^c+\gamma)} \prod_{j=1}^{n} \left(t_j^c+\gamma\right)^{-1} \\ &\propto c^d \gamma^n e^{W_1(\gamma,c)-W_2(\gamma,c)},\end{aligned}$$

where the functions W_1 and W_2 are written in log form and defined as

$$W_1(\gamma, c) = \sum_{i=1}^{d} \left[(c-1)\ln t_i - \ln(t_i^c+\gamma)\right] \tag{5.5}$$

$$\text{and } W_2(\gamma, c) = \sum_{i=1}^{n} \ln\left(t_i^c+\gamma\right). \tag{5.6}$$

The proportional form of the posterior distribution of γ and c corresponding to the Jeffreys prior (5.4) can now be constructed, such that

$$\pi_{\text{jeff}}(\gamma, c|\mathbf{t}) \propto \pi_{\text{jeff}}(\gamma, c) \cdot \mathcal{L}(\gamma, c|\mathbf{t}). \tag{5.7}$$

5.1.3 Bayesian estimators of the parameters

The posterior distribution of the parameters is required to formally define the Bayesian estimators of interest, special cases of those derived in Section 2.2.2.3. Only the Jeffreys prior is considered, so there is only one set of estimators for each of the loss functions, corresponding to the posterior (5.7). Under the symmetric loss functions, AE and SE, the estimators become the posterior expected values and medians, such that

$$\begin{aligned}(\hat{\gamma}_{\text{AE}},\ \hat{c}_{\text{AE}}) &= \text{median}_{\gamma,c|\mathbf{t}}[(\gamma, c)] \\ (\hat{\gamma}_{\text{SE}},\ \hat{c}_{\text{SE}}) &= \text{E}_{\gamma,c|\mathbf{t}}[(\gamma, c)].\end{aligned}$$

Under the asymmetric LINEX loss function with its parameter a and the GE loss function with its parameter k, the Bayesian estimators are given by

$$\begin{aligned}\left(\hat{\gamma}_{\text{LNX(a)}},\ \hat{c}_{\text{LNX(a)}}\right) &= -\frac{1}{a}\ln \mathrm{E}_{\gamma,c|\mathbf{t}}\left[e^{-a(\gamma,c)}\right] \\ \left(\hat{\gamma}_{\text{GE(k)}},\ \hat{c}_{\text{GE(k)}}\right) &= \left(\mathrm{E}_{\gamma,c|\mathbf{t}}\left[(\gamma,c)^{-k}\right]\right)^{-\frac{1}{k}}.\end{aligned}$$

The derivation of Bayesian estimators for functions of the parameters of interest yield analogous results to those above, but only the hazard rate (5.1) is shown here. Its estimators, under the different loss functions, become

$$\begin{aligned}\hat{h}_{\text{SE}}(t,\gamma,c) &= \mathrm{E}_{\gamma,c|\mathbf{t}}[h(t,\gamma,c)] \\ \hat{h}_{\text{AE}}(t,\gamma,c) &= \text{median}_{\gamma,c|\mathbf{t}}[h(t,\gamma,c)] \\ \hat{h}_{\text{LNX(a)}}(t,\gamma,c) &= -\frac{1}{a}\ln \mathrm{E}_{\gamma,c|\mathbf{t}}\left[e^{-ah(t,\gamma,c)}\right] \\ \hat{h}_{\text{GE(k)}}(t,\gamma,c) &= \left(\mathrm{E}_{\gamma,c|\mathbf{t}}\left[h(t,\gamma,c)^{-k}\right]\right)^{-\frac{1}{k}}.\end{aligned}$$

5.1.4 Simulation results

A simulation study was performed to assess the performance of the Bayesian estimators for the GCRE model parameters, as described in Section 3.3.2. The process was carried out for five different pairs of parameter values:

$$\begin{aligned}\alpha &= 0.5\ , \qquad c = 3.0 \\ \alpha &= 1.0\ , \qquad c = 4.0 \\ \alpha &= 0.5\ , \qquad c = 4.0 \\ \alpha &= 1.0\ , \qquad c = 3.0 \\ \alpha &= 0.5\ , \qquad c = 5.0\end{aligned}$$

as well as two levels of censoring: $\delta = \{1, 0.8\}$. As in Chapter 4, one of these pairs of parameter values is chosen to investigate the effect of sample size on the estimators. Simulations are run for ($\alpha = 1.0$, $c = 4.0$) using sample sizes of both 30 and 50, whereas for the remaining parameter value pairs, only $n = 50$ is used.

The simulation results are shown predominantly in the form of plots, which is described in detail in Section 4.1.4. Figures 5.2, 5.5, 5.11, 5.14 and 5.17 show the Bayesian point estimates of the model parameters, while Figures 5.4, 5.7, 5.13, 5.16 and 5.19 portray estimates of the hazard rate curves. Coverages and MSE vs bias are plotted in Figures 5.3, 5.6, 5.12, 5.15 and 5.18, while the MAE, MSE and bias of all estimators are summarised in Tables 5.1, 5.2, 5.4, 5.5 and 5.6. The simulation results with a reduced sample size of $n = 30$ are shown in Figures 5.8, 5.10 and 5.9 and Table 5.3.

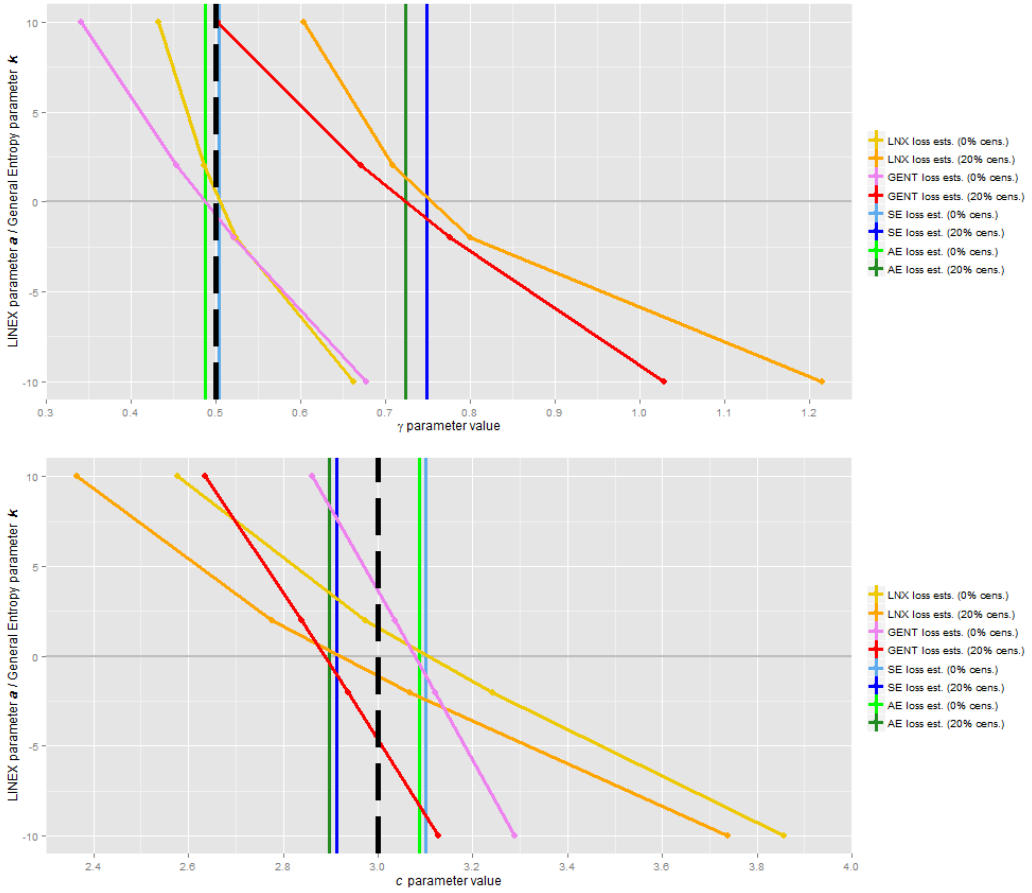


Figure 5.2: *Bayesian estimates plot for GCRE model, with* $\gamma = 0.5$ *(top) and* $c = 3$ *(bottom), and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours).*

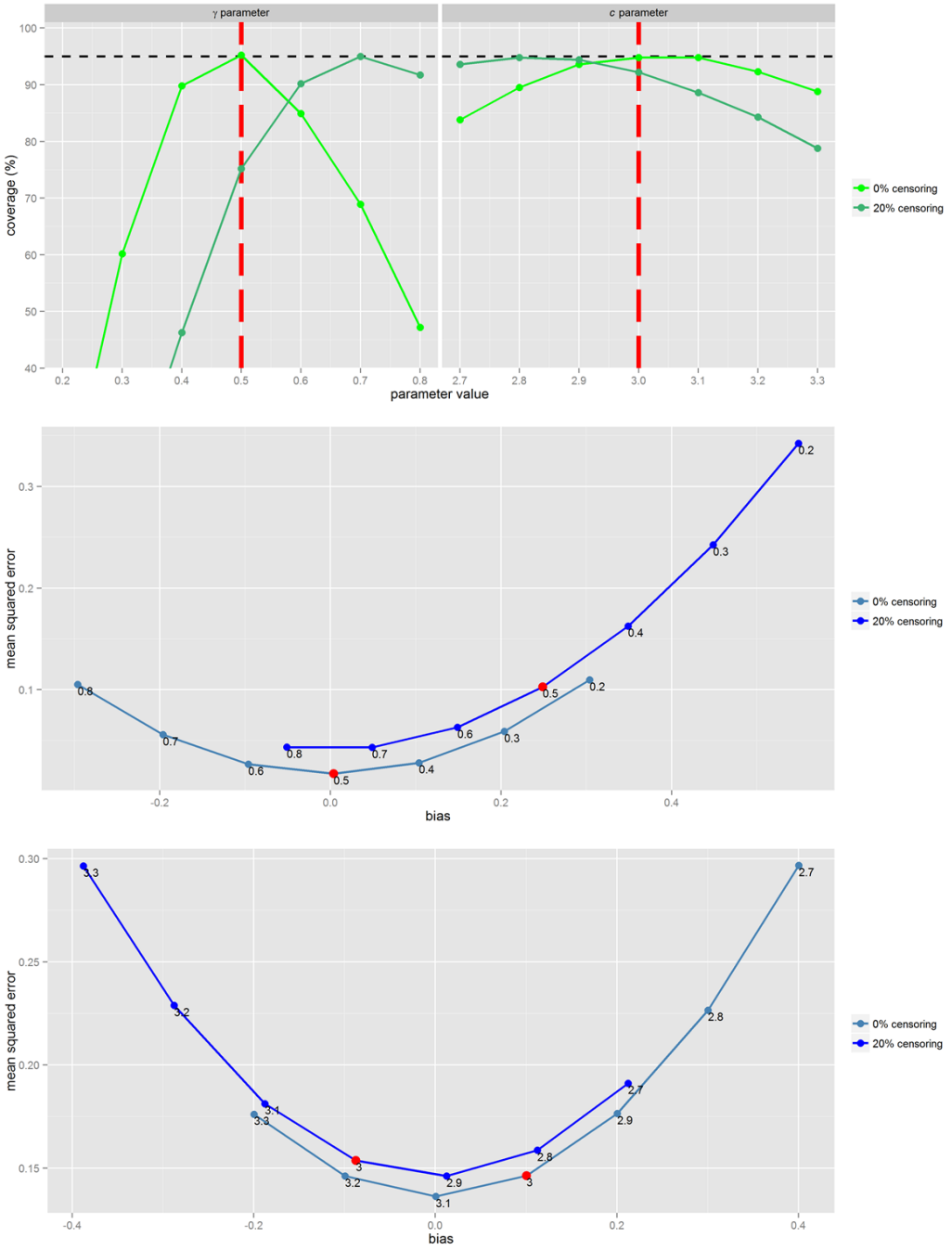


Figure 5.3: *Coverage plots (top) as well as MSE vs bias plots (bottom two) for GCRE model, with* $\gamma = 0.5$ *and* $c = 3$*, and* $\delta = 1$ *(no censoring, top) and* $\delta = 0.8$ *(20% censored values, bottom).*

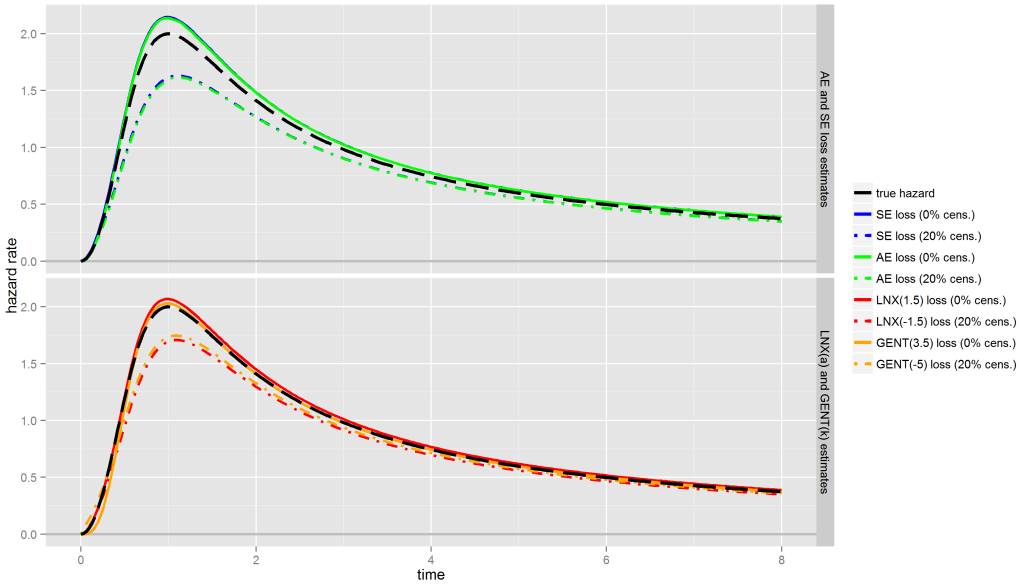


Figure 5.4: *Plots of Bayesian estimates of the hazard function for the GCRE model with* $(\gamma, c) = (0.5, 3)$ *and two levels of censoring, derived using four different loss functions.*

Table 5.1: *The MAE, MSE and bias of estimators for the GCRE model, with parameters* $(\gamma, c) = (0.5, 3)$.

estimator	MAE	MSE	bias	estimator MAE	MSE	bias	
0% censoring							
$\hat{\gamma}_{\text{AE}}$	0.1043	0.0167	-0.0122	$\hat{c}_{\text{AE}}$	0.2969	0.1426	0.0871
$\hat{\gamma}_{\text{SE}}$	0.1049	0.0174	0.0038	$\hat{c}_{\text{SE}}$	0.3001	0.1464	0.1004
$\hat{\gamma}_{\text{LNX(-2)}}$	0.1126	0.0209	0.0238	$\hat{c}_{\text{LNX(-2)}}$	0.3597	0.2217	0.262
$\hat{\gamma}_{\text{LNX(2)}}$	0.1014	0.0158	-0.0143	$\hat{c}_{\text{LNX(2)}}$	0.2525	0.106	-0.0449
$\hat{\gamma}_{\text{GE(-2)}}$	0.1078	0.019	0.0209	$\hat{c}_{\text{GE(-2)}}$	0.2928	0.1468	0.1211
$\hat{\gamma}_{\text{GE(2)}}$	0.1085	0.0173	-0.0469	$\hat{c}_{\text{GE(2)}}$	0.2704	0.1233	0.0256
20% censoring							
$\hat{\gamma}_{\text{AE}}$	0.2388	0.0883	0.2236	$\hat{c}_{\text{AE}}$	0.3156	0.1547	-0.1029
$\hat{\gamma}_{\text{SE}}$	0.2606	0.1029	0.2489	$\hat{c}_{\text{SE}}$	0.3137	0.1538	-0.0877
$\hat{\gamma}_{\text{LNX(-2)}}$	0.3094	0.1438	0.2996	$\hat{c}_{\text{LNX(-2)}}$	0.339	0.1931	0.0642
$\hat{\gamma}_{\text{LNX(2)}}$	0.2246	0.0775	0.2085	$\hat{c}_{\text{LNX(2)}}$	0.3422	0.1803	-0.227
$\hat{\gamma}_{\text{GE(-2)}}$	0.2852	0.12	0.2755	$\hat{c}_{\text{GE(-2)}}$	0.3136	0.1613	-0.0662
$\hat{\gamma}_{\text{GE(2)}}$	0.1974	0.0637	0.1708	$\hat{c}_{\text{GE(2)}}$	0.3315	0.1741	-0.1644

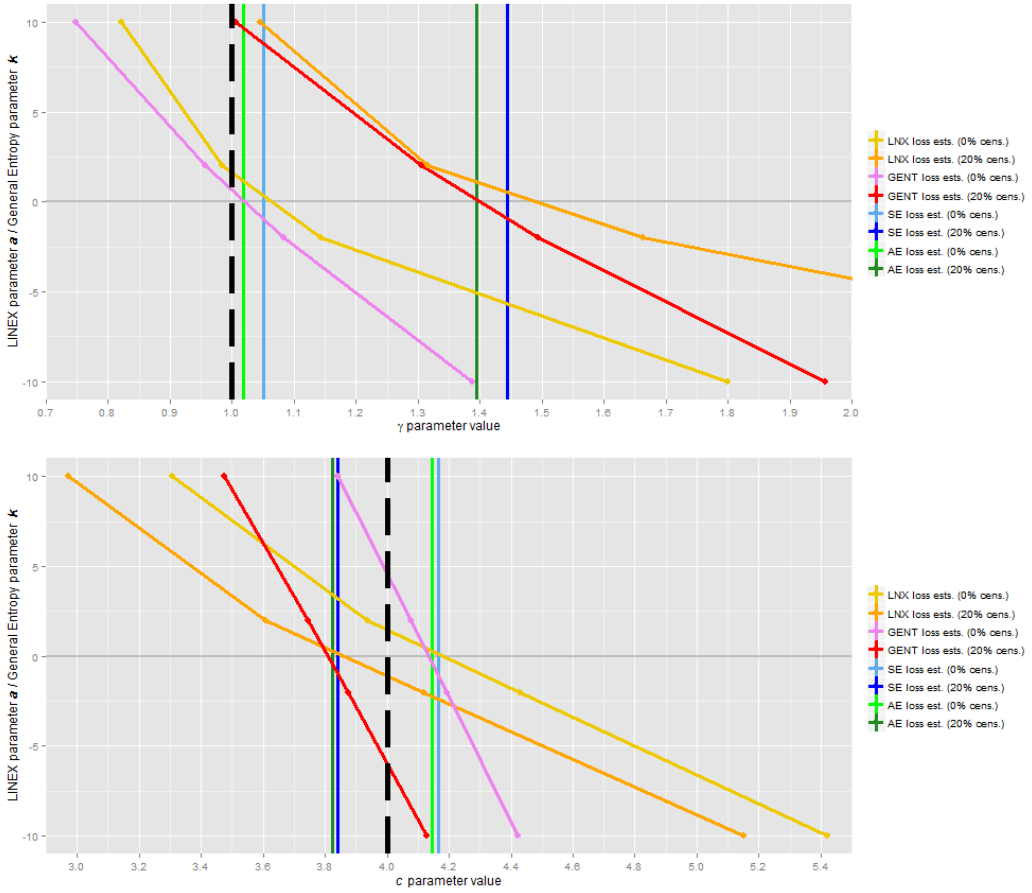


Figure 5.5: *Bayesian estimates plot for GCRE model, with* $\gamma = 1$ *(top) and* $c = 4$ *(bottom), and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours).*

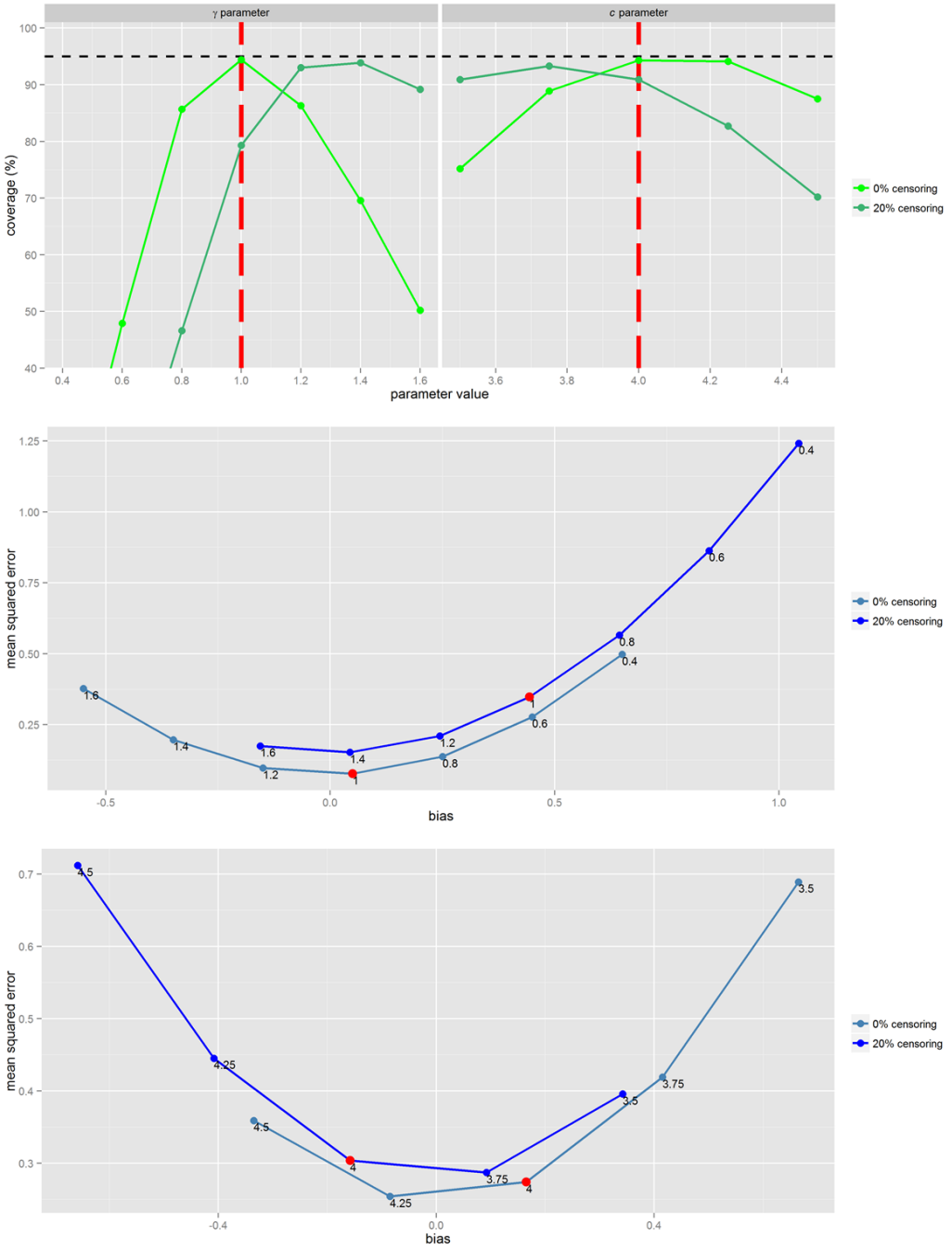


Figure 5.6: *Coverage plots (top) as well as MSE vs bias plots (bottom two) for GCRE model, with* $\gamma = 1$ *and* $c = 4$*, and* $\delta = 1$ *(no censoring, top) and* $\delta = 0.8$ *(20% censored values, bottom).*

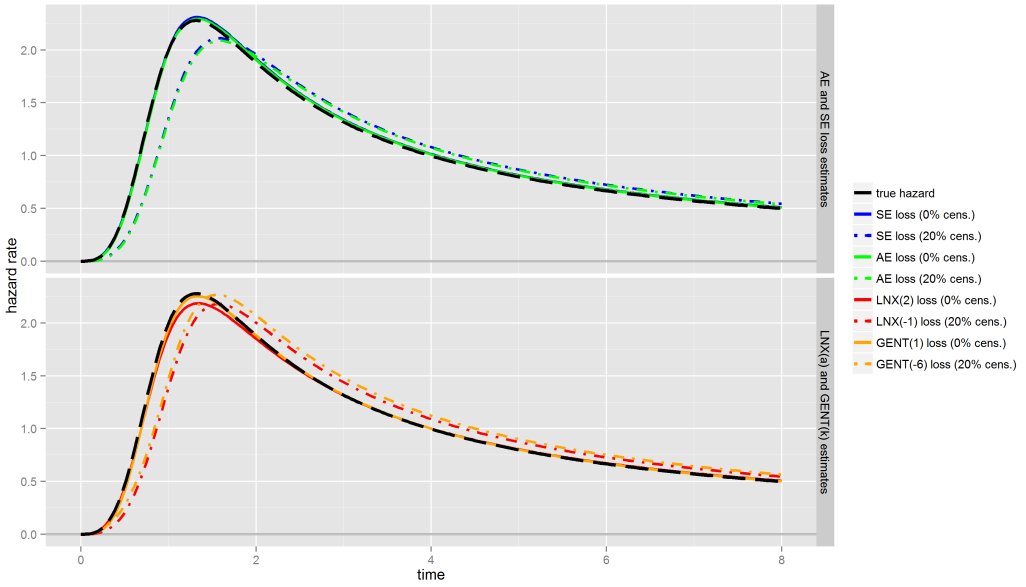


Figure 5.7: *Plots of Bayesian estimates of the hazard function for the GCRE model with* $(\gamma, c) = (1, 4)$ *and two levels of censoring, derived using four different loss functions.*

Table 5.2: *The MAE, MSE and bias of estimators for the GCRE model, with parameters* $(\gamma, c) = (1, 4)$.

estimator	MAE	MSE	bias	estimator	MAE	MSE	bias
0% censoring							
$\hat{\gamma}_{\text{AE}}$	0.206	0.0704	0.0182	$\hat{c}_{\text{AE}}$	0.3987	0.2658	0.1467
$\hat{\gamma}_{\text{SE}}$	0.2132	0.0774	0.0501	$\hat{c}_{\text{SE}}$	0.4043	0.2741	0.165
$\hat{\gamma}_{\text{LNX(-2)}}$	0.2669	0.1327	0.1414	$\hat{c}_{\text{LNX(-2)}}$	0.5485	0.5191	0.4251
$\hat{\gamma}_{\text{LNX(2)}}$	0.1925	0.0592	-0.0164	$\hat{c}_{\text{LNX(2)}}$	0.3629	0.2163	-0.0662
$\hat{\gamma}_{\text{GE(-2)}}$	0.2241	0.0876	0.0821	$\hat{c}_{\text{GE(-2)}}$	0.4168	0.3037	0.1896
$\hat{\gamma}_{\text{GE(2)}}$	0.2047	0.0652	-0.0442	$\hat{c}_{\text{GE(2)}}$	0.3893	0.2583	0.0739
20% censoring							
$\hat{\gamma}_{\text{AE}}$	0.4238	0.2946	0.3947	$\hat{c}_{\text{AE}}$	0.4507	0.3074	-0.1778
$\hat{\gamma}_{\text{SE}}$	0.4669	0.3481	0.444	$\hat{c}_{\text{SE}}$	0.4459	0.3037	-0.1581
$\hat{\gamma}_{\text{LNX(-2)}}$	0.6772	0.7408	0.6617	$\hat{c}_{\text{LNX(-2)}}$	0.4759	0.4007	0.1153
$\hat{\gamma}_{\text{LNX(2)}}$	0.3517	0.2021	0.3145	$\hat{c}_{\text{LNX(2)}}$	0.5178	0.388	-0.3961
$\hat{\gamma}_{\text{GE(-2)}}$	0.5118	0.4073	0.4922	$\hat{c}_{\text{GE(-2)}}$	0.4428	0.3146	-0.1296
$\hat{\gamma}_{\text{GE(2)}}$	0.3528	0.2147	0.3043	$\hat{c}_{\text{GE(2)}}$	0.4776	0.3469	-0.2597

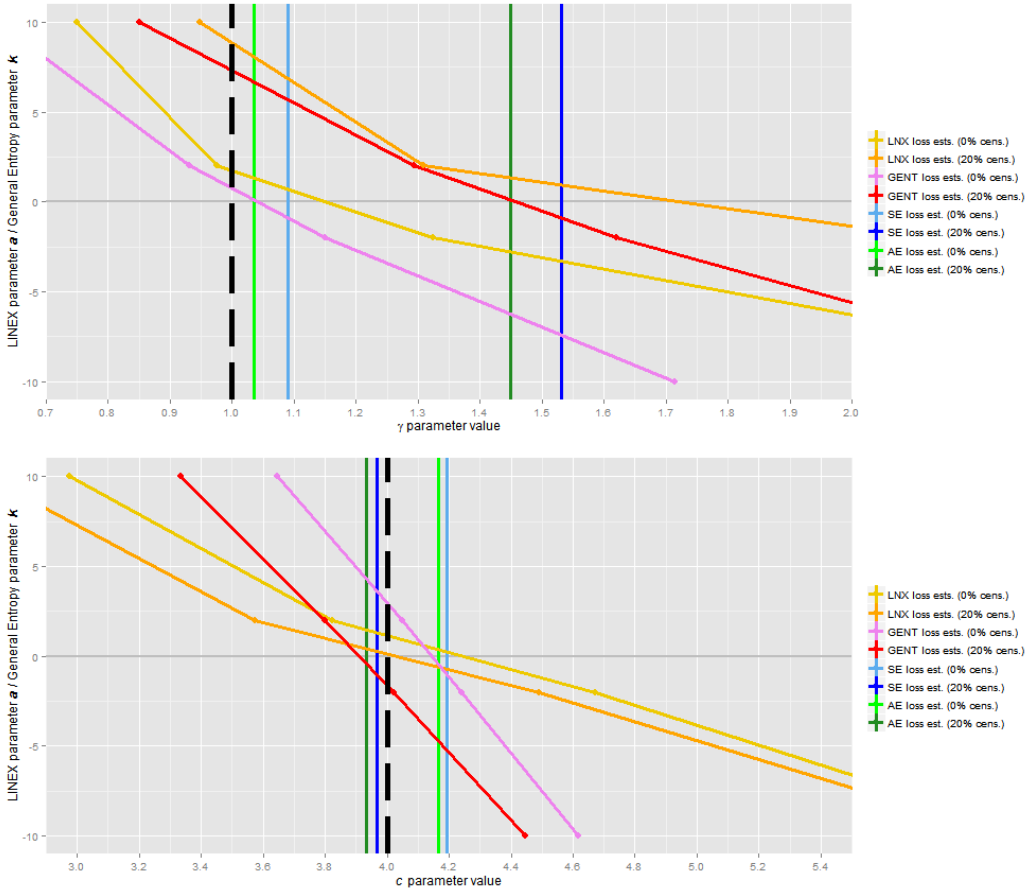


Figure 5.8: *Bayesian estimates plot for GCRE model, with* $\gamma = 1$ *(top) and* $c = 4$ *(bottom), and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours), and with* $n = 30$.

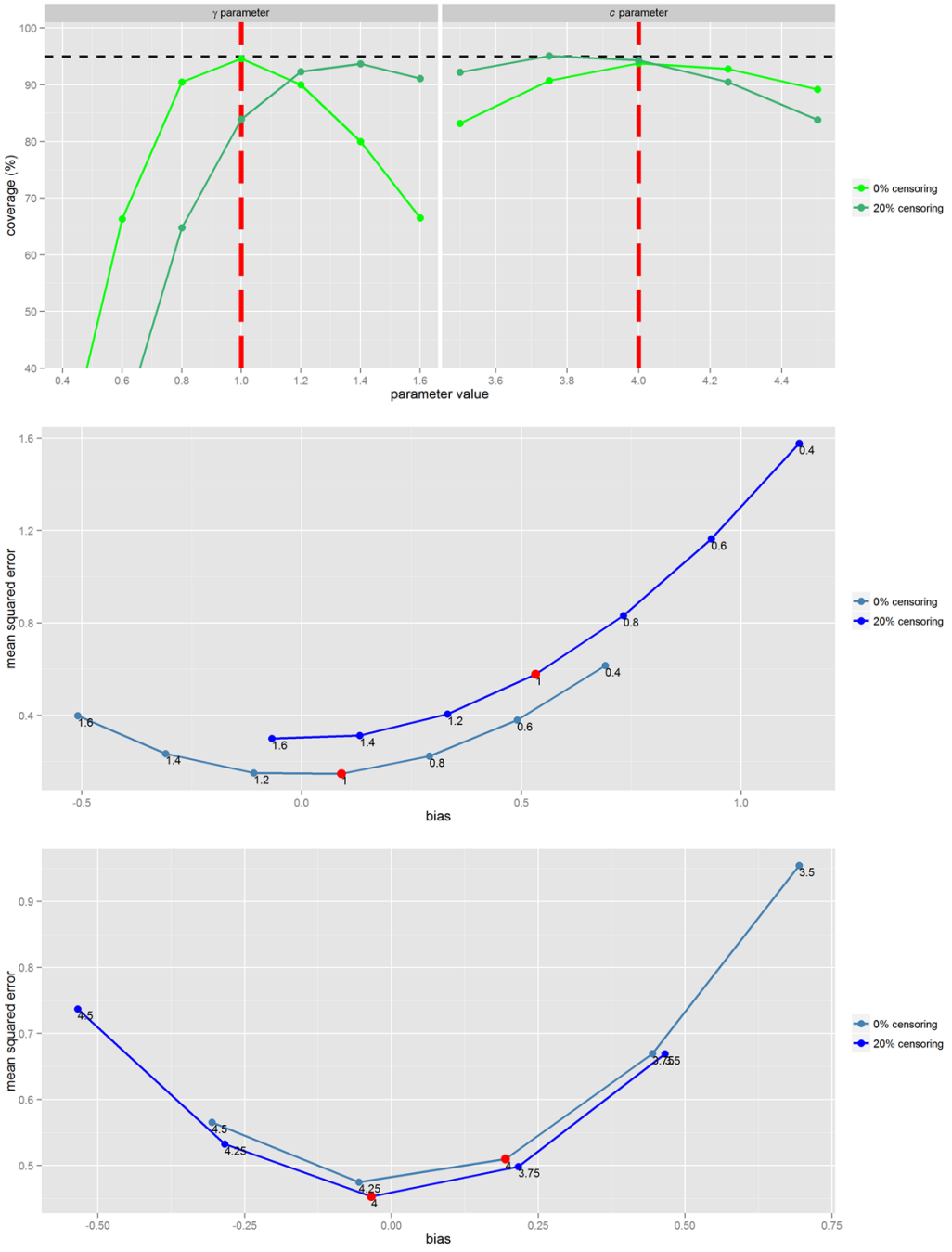


Figure 5.9: *Coverage plots (top) as well as MSE vs bias plots (bottom two) for GCRE model, with* $\gamma = 1$ *and* $c = 4$*, and* $\delta = 1$ *(no censoring, top) and* $\delta = 0.8$ *(20% censored values, bottom), and with* $n = 30$*.*

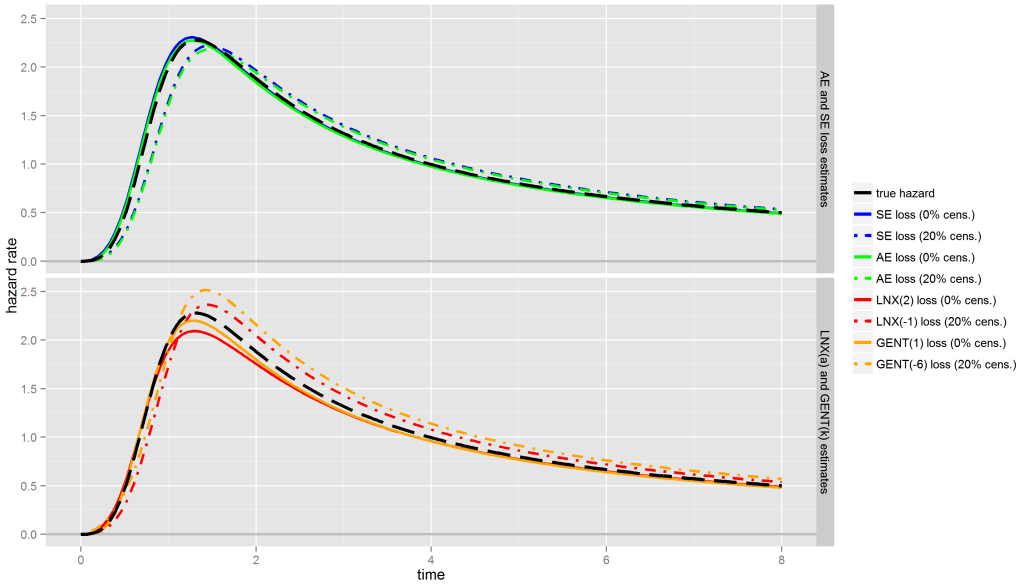


Figure 5.10: *Plots of Bayesian estimates of the hazard function for the GCRE model with* $(\gamma, c) = (1, 4)$ *and two levels of censoring, derived using four different loss functions and with sample size* $n = 30$.

Table 5.3: *The MAE, MSE and bias of estimators for the GCRE model, with parameters* $(\gamma, c) = (1, 4)$ *and* $n = 30$.

estimator	MAE	MSE	bias	estimator	MAE	MSE	bias
0% censoring							
$\hat{\gamma}_{\text{AE}}$	0.2677	0.126	0.0357	$\hat{c}_{\text{AE}}$	0.5302	0.4924	0.1641
$\hat{\gamma}_{\text{SE}}$	0.285	0.1472	0.0906	$\hat{c}_{\text{SE}}$	0.5369	0.51	0.1944
$\hat{\gamma}_{\text{LNX(-2)}}$	0.4592	0.5382	0.3231	$\hat{c}_{\text{LNX(-2)}}$	0.8157	1.2163	0.6681
$\hat{\gamma}_{\text{LNX(2)}}$	0.2364	0.0912	-0.025	$\hat{c}_{\text{LNX(2)}}$	0.4885	0.3673	-0.1798
$\hat{\gamma}_{\text{GE(-2)}}$	0.3096	0.1774	0.148	$\hat{c}_{\text{GE(-2)}}$	0.5534	0.5576	0.2382
$\hat{\gamma}_{\text{GE(2)}}$	0.2597	0.1079	-0.0696	$\hat{c}_{\text{GE(2)}}$	0.5179	0.4567	0.0444
20% censoring							
$\hat{\gamma}_{\text{AE}}$	0.5074	0.4665	0.4486	$\hat{c}_{\text{AE}}$	0.5393	0.4502	-0.0684
$\hat{\gamma}_{\text{SE}}$	0.5772	0.5784	0.532	$\hat{c}_{\text{SE}}$	0.5377	0.4533	-0.0341
$\hat{\gamma}_{\text{LNX(-2)}}$	1.1538	2.7127	1.1271	$\hat{c}_{\text{LNX(-2)}}$	0.7488	1.009	0.4875
$\hat{\gamma}_{\text{LNX(2)}}$	0.3807	0.255	0.3076	$\hat{c}_{\text{LNX(2)}}$	0.5894	0.4994	-0.4312
$\hat{\gamma}_{\text{GE(-2)}}$	0.6542	0.7152	0.6187	$\hat{c}_{\text{GE(-2)}}$	0.5416	0.48	0.0167
$\hat{\gamma}_{\text{GE(2)}}$	0.3947	0.2967	0.293	$\hat{c}_{\text{GE(2)}}$	0.5586	0.473	-0.2049

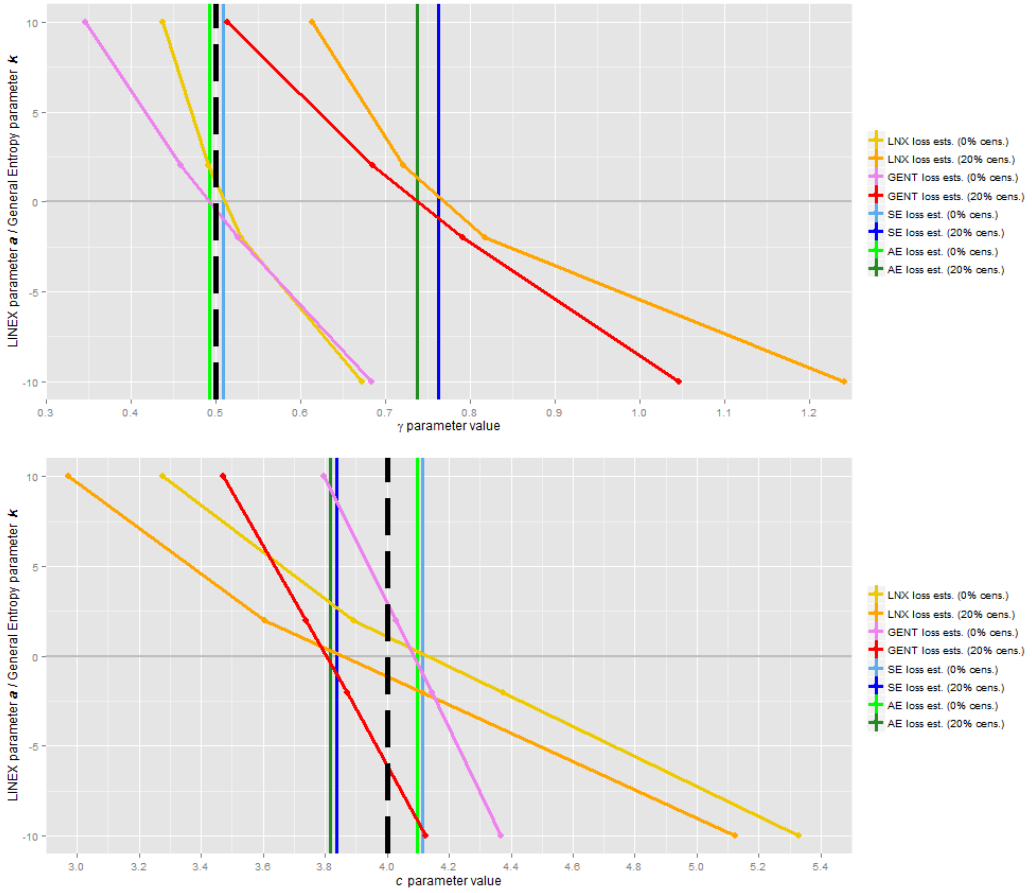


Figure 5.11: *Bayesian estimates plot for GCRE model, with* $\gamma = 0.5$ *(top) and* $c = 4$ *(bottom), and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours).*

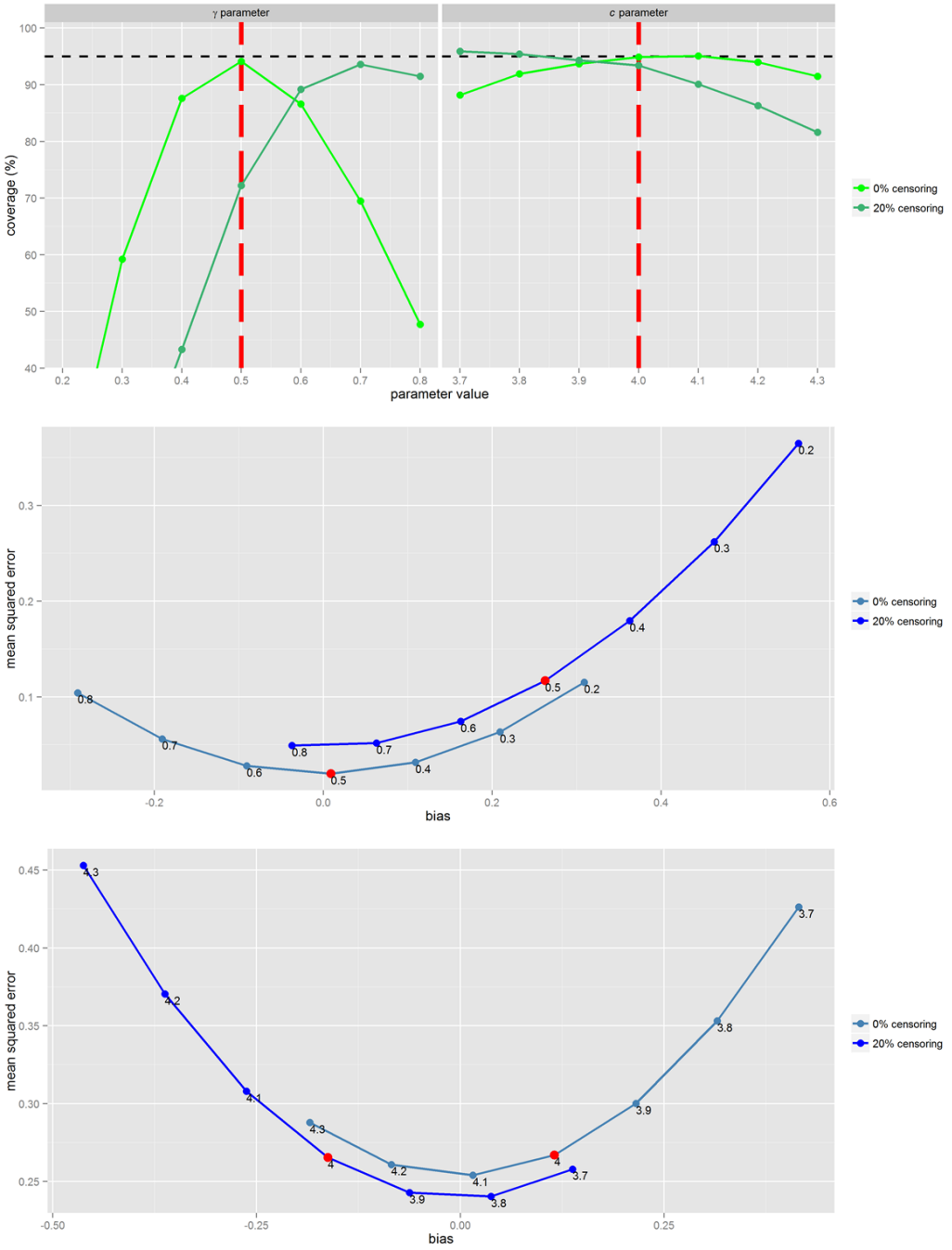


Figure 5.12: *Coverage plots (top) as well as MSE vs bias plots (bottom two) for GCRE model, with* $\gamma = 0.5$ *and* $c = 4$*, and* $\delta = 1$ *(no censoring, top) and* $\delta = 0.8$ *(20% censored values, bottom).*

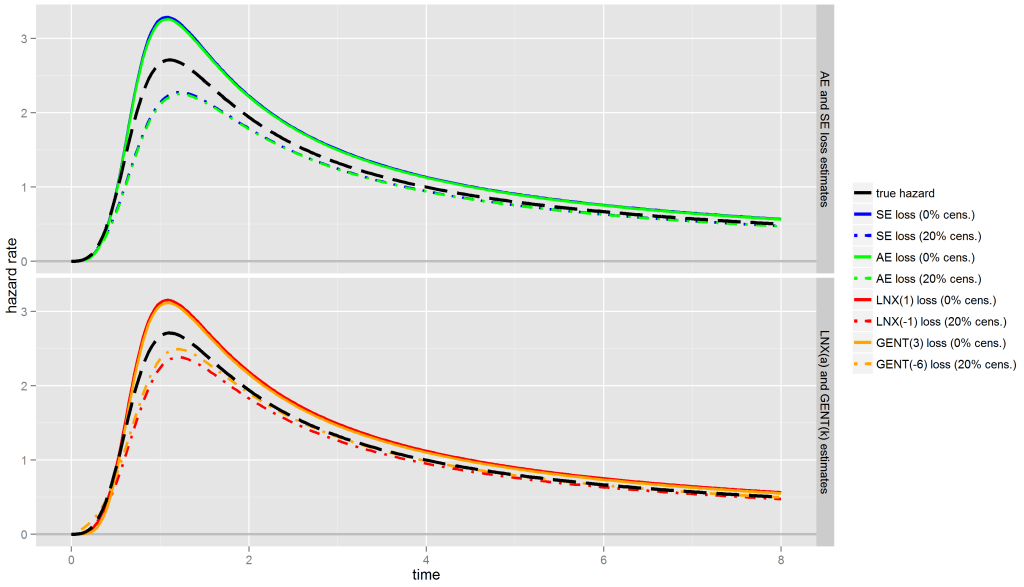


Figure 5.13: *Plots of Bayesian estimates of the hazard function for the GCRE model with* $(\gamma, c) = (0.5, 4)$ *and two levels of censoring, derived using four different loss functions.*

Table 5.4: *The MAE, MSE and bias of estimators for the GCRE model, with parameters* $(\gamma, c) = (0.5, 4)$.

estimator	MAE	MSE	bias	estimator	MAE	MSE	bias
0% censoring							
$\hat{\gamma}_{\text{AE}}$	0.1084	0.0188	-0.0069	$\hat{c}_{\text{AE}}$	0.4016	0.2609	0.0973
$\hat{\gamma}_{\text{SE}}$	0.1091	0.0198	0.0091	$\hat{c}_{\text{SE}}$	0.4047	0.2671	0.1153
$\hat{\gamma}_{\text{LNX(-2)}}$	0.1172	0.024	0.0296	$\hat{c}_{\text{LNX(-2)}}$	0.5243	0.4808	0.3688
$\hat{\gamma}_{\text{LNX(2)}}$	0.1054	0.0177	-0.0094	$\hat{c}_{\text{LNX(2)}}$	0.3789	0.2296	-0.1112
$\hat{\gamma}_{\text{GE(-2)}}$	0.1121	0.0216	0.0263	$\hat{c}_{\text{GE(-2)}}$	0.4148	0.2937	0.1397
$\hat{\gamma}_{\text{GE(2)}}$	0.1123	0.0189	-0.0419	$\hat{c}_{\text{GE(2)}}$	0.3964	0.2597	0.0252
20% censoring							
$\hat{\gamma}_{\text{AE}}$	0.2518	0.1009	0.2369	$\hat{c}_{\text{AE}}$	0.4219	0.27	-0.1829
$\hat{\gamma}_{\text{SE}}$	0.2743	0.1169	0.2628	$\hat{c}_{\text{SE}}$	0.4174	0.2654	-0.1624
$\hat{\gamma}_{\text{LNX(-2)}}$	0.3257	0.1649	0.3161	$\hat{c}_{\text{LNX(-2)}}$	0.4513	0.3429	0.1095
$\hat{\gamma}_{\text{LNX(2)}}$	0.2366	0.0882	0.221	$\hat{c}_{\text{LNX(2)}}$	0.4939	0.3608	-0.3997
$\hat{\gamma}_{\text{GE(-2)}}$	0.2993	0.1355	0.2898	$\hat{c}_{\text{GE(-2)}}$	0.415	0.2754	-0.1339
$\hat{\gamma}_{\text{GE(2)}}$	0.2097	0.0741	0.1838	$\hat{c}_{\text{GE(2)}}$	0.4491	0.3115	-0.2641

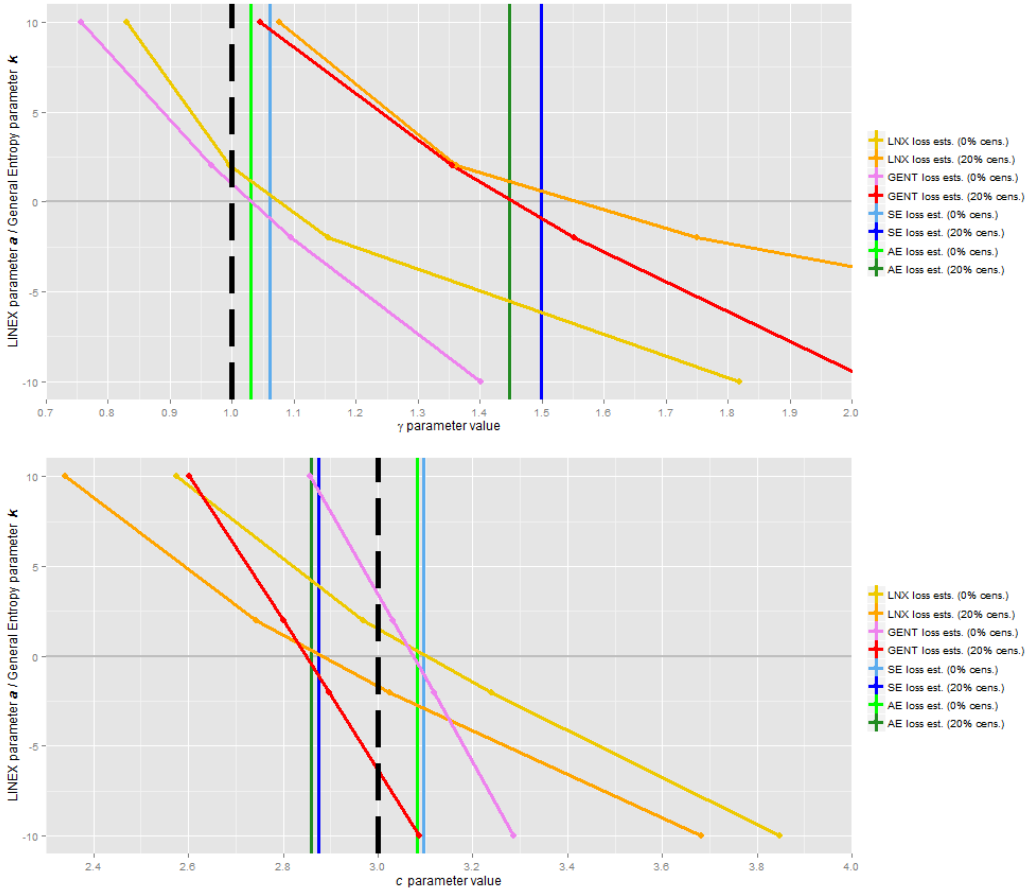


Figure 5.14: *Bayesian estimates plot for GCRE model, with* $\gamma = 1$ *(top) and* $c = 3$ *(bottom), and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours).*

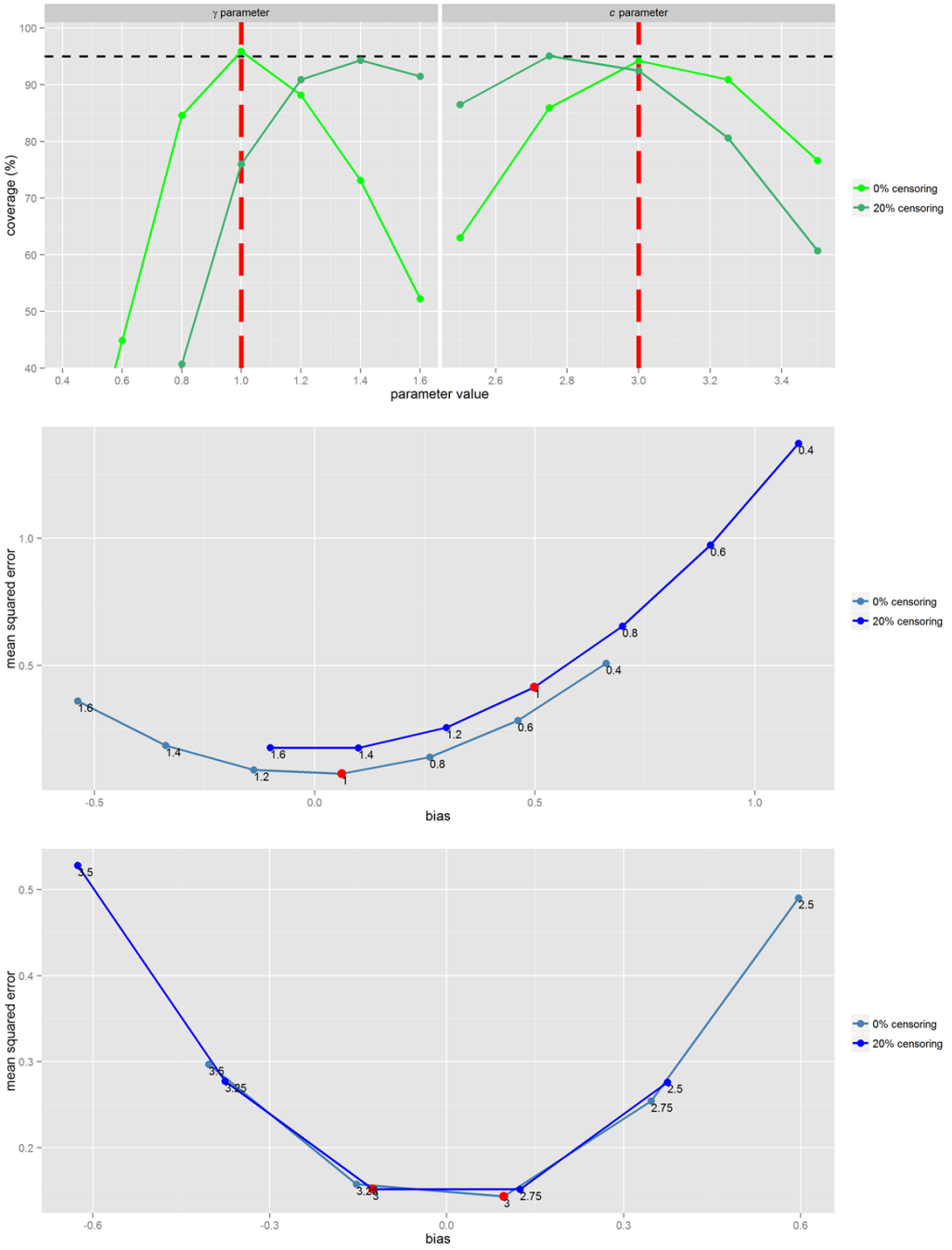


Figure 5.15: *Coverage plots (top) as well as MSE vs bias plots (bottom two) for GCRE model, with* $\gamma = 1$ *and* $c = 3$*, and* $\delta = 1$ *(no censoring, top) and* $\delta = 0.8$ *(20% censored values, bottom).*

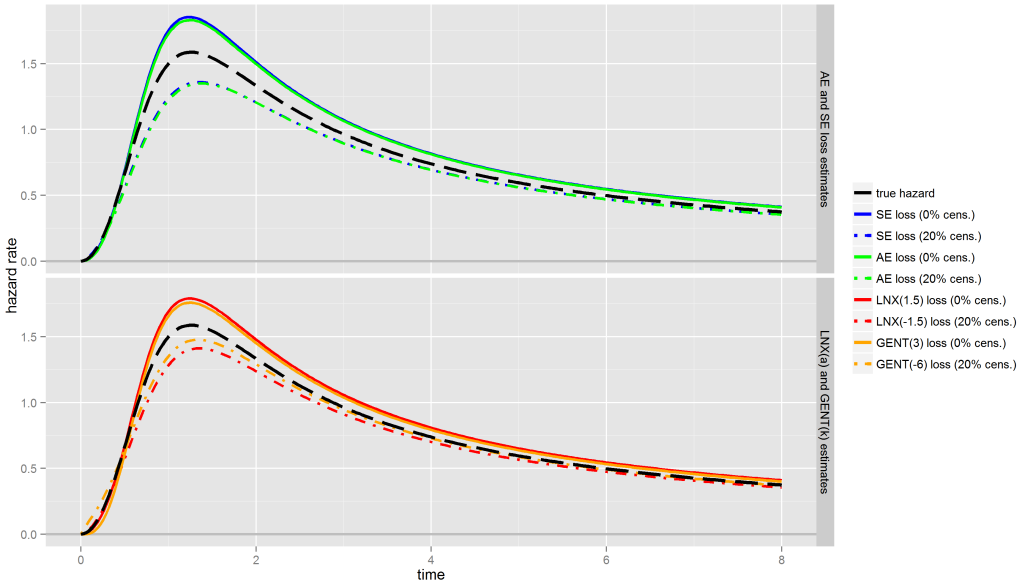


Figure 5.16: *Plots of Bayesian estimates of the hazard function for the GCRE model with* $(\gamma, c) = (1, 3)$ *and two levels of censoring, derived using four different loss functions.*

Table 5.5: *The MAE, MSE and bias of estimators for the GCRE model, with parameters* $(\gamma, c) = (1, 3)$.

estimator	MAE	MSE	bias	estimator	MAE	MSE	bias
0% censoring							
$\hat{\gamma}_{\text{AE}}$	0.2009	0.0662	0.0294	$\hat{c}_{\text{AE}}$	0.2947	0.14	0.0836
$\hat{\gamma}_{\text{SE}}$	0.2096	0.0738	0.0617	$\hat{c}_{\text{SE}}$	0.2979	0.1436	0.0967
$\hat{\gamma}_{\text{LNX(-2)}}$	0.2652	0.1272	0.1537	$\hat{c}_{\text{LNX(-2)}}$	0.3652	0.227	0.2352
$\hat{\gamma}_{\text{LNX(2)}}$	0.1862	0.0552	-0.0058	$\hat{c}_{\text{LNX(2)}}$	0.2755	0.1231	-0.0335
$\hat{\gamma}_{\text{GE(-2)}}$	0.222	0.0845	0.094	$\hat{c}_{\text{GE(-2)}}$	0.3064	0.1589	0.1148
$\hat{\gamma}_{\text{GE(2)}}$	0.1965	0.0603	-0.0334	$\hat{c}_{\text{GE(2)}}$	0.2892	0.1385	0.0294
20% censoring							
$\hat{\gamma}_{\text{AE}}$	0.4689	0.3535	0.4478	$\hat{c}_{\text{AE}}$	0.3153	0.1547	-0.1406
$\hat{\gamma}_{\text{SE}}$	0.5156	0.4147	0.4991	$\hat{c}_{\text{SE}}$	0.3109	0.1521	-0.1258
$\hat{\gamma}_{\text{LNX(-2)}}$	0.7592	1.0085	0.7478	$\hat{c}_{\text{LNX(-2)}}$	0.3176	0.1758	0.0208
$\hat{\gamma}_{\text{LNX(2)}}$	0.3873	0.2403	0.3598	$\hat{c}_{\text{LNX(2)}}$	0.3571	0.1891	-0.2609
$\hat{\gamma}_{\text{GE(-2)}}$	0.5638	0.4831	0.5494	$\hat{c}_{\text{GE(-2)}}$	0.3078	0.1576	-0.1047
$\hat{\gamma}_{\text{GE(2)}}$	0.3902	0.2585	0.3534	$\hat{c}_{\text{GE(2)}}$	0.3392	0.1782	-0.2012

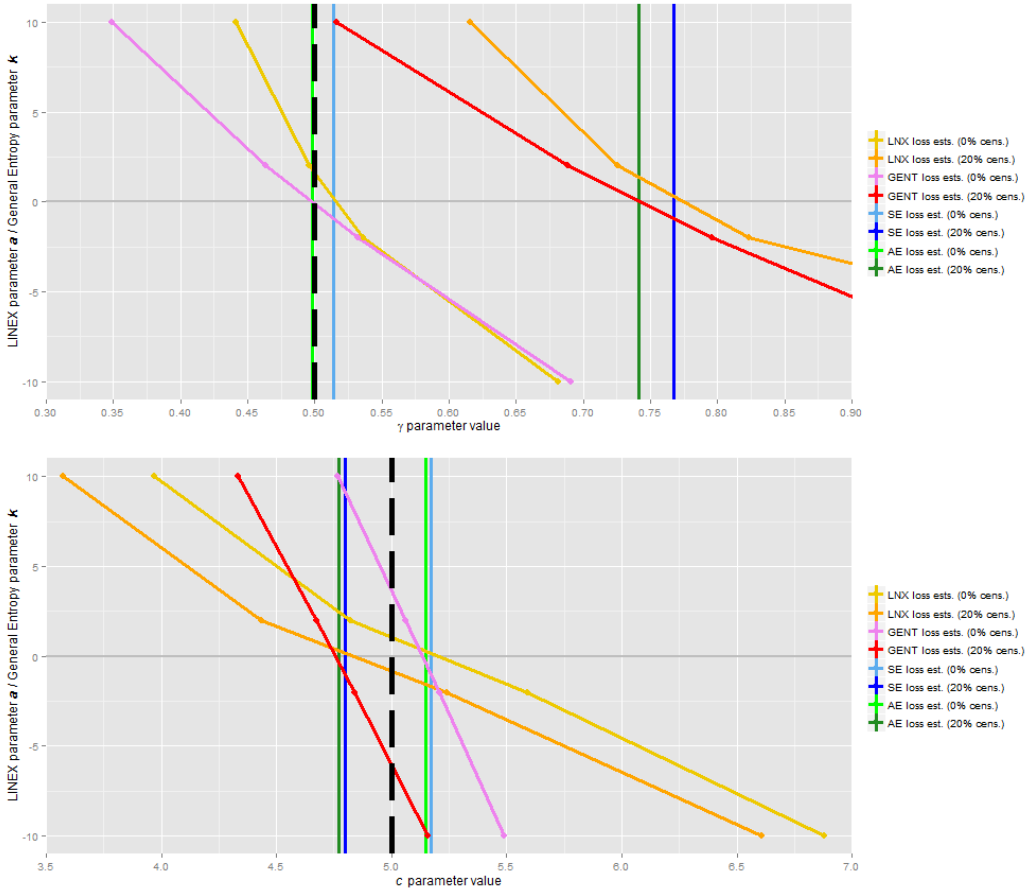


Figure 5.17: *Bayesian estimates plot for GCRE model, with* $\gamma = 0.5$ *(top) and* $c = 5$ *(bottom), and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours).*

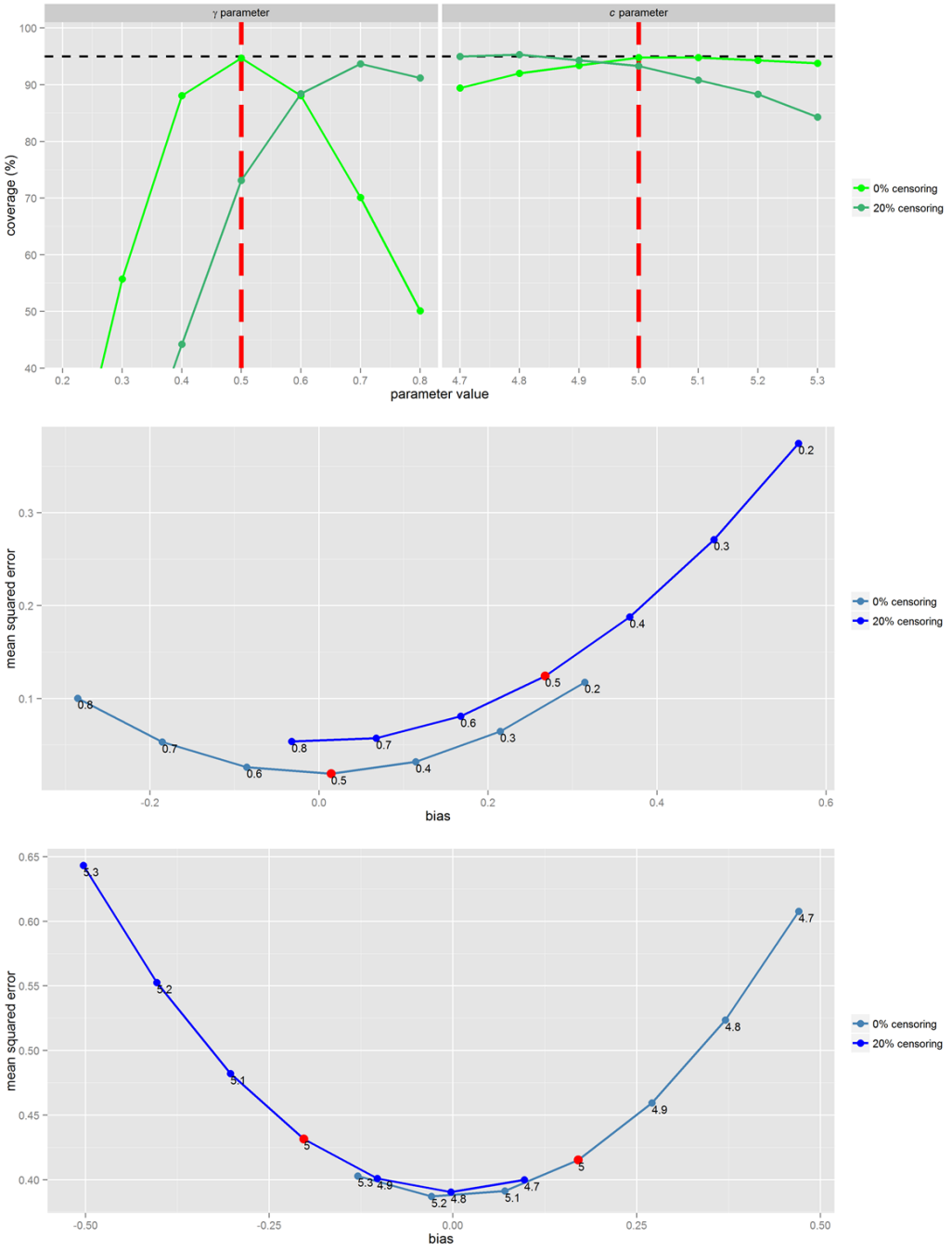


Figure 5.18: *Coverage plots (top) as well as MSE vs bias plots (bottom two) for GCRE model, with* $\gamma = 0.5$ *and* $c = 5$*, and* $\delta = 1$ *(no censoring, top) and* $\delta = 0.8$ *(20% censored values, bottom).*

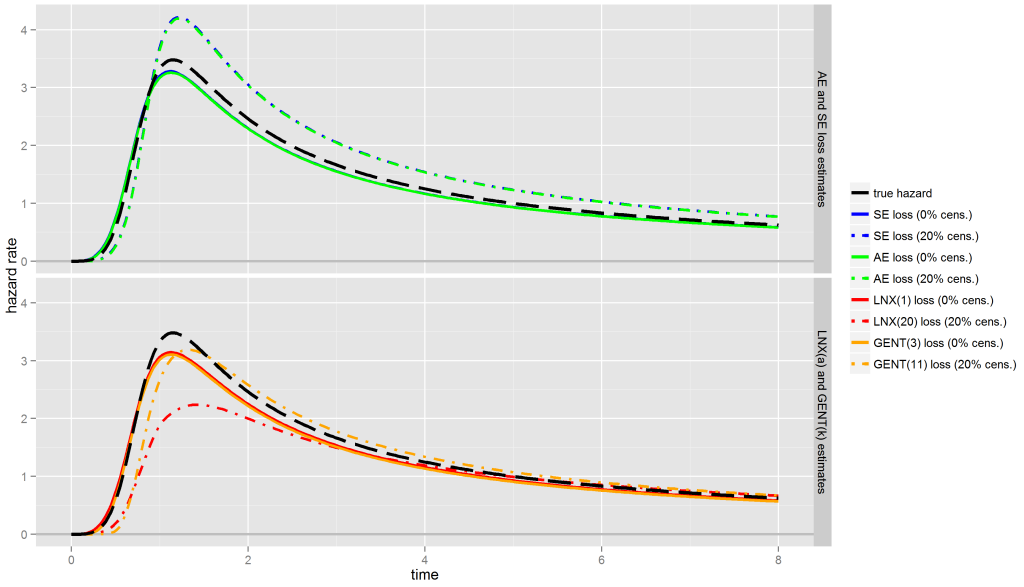


Figure 5.19: *Plots of Bayesian estimates of the hazard function for the GCRE model with* $(\gamma, c) = (0.5, 5)$ *and two levels of censoring, derived using four different loss functions.*

Table 5.6: *The MAE, MSE and bias of estimators for the GCRE model, with parameters* $(\gamma, c) = (0.5, 5)$.

estimator	MAE	MSE	bias	estimator	MAE	MSE	bias
0% censoring							
$\hat{\gamma}_{\text{AE}}$	0.1062	0.0178	-0.002	$\hat{c}_{\text{AE}}$	0.4983	0.4047	0.1479
$\hat{\gamma}_{\text{SE}}$	0.1081	0.0189	0.0143	$\hat{c}_{\text{SE}}$	0.5033	0.4153	0.1706
$\hat{\gamma}_{\text{LNX(-2)}}$	0.117	0.0232	0.0352	$\hat{c}_{\text{LNX(-2)}}$	0.7296	0.907	0.5836
$\hat{\gamma}_{\text{LNX(2)}}$	0.103	0.0168	-0.0045	$\hat{c}_{\text{LNX(2)}}$	0.468	0.3459	-0.1811
$\hat{\gamma}_{\text{GE(-2)}}$	0.1119	0.021	0.0317	$\hat{c}_{\text{GE(-2)}}$	0.5169	0.459	0.2014
$\hat{\gamma}_{\text{GE(2)}}$	0.1082	0.0177	-0.0371	$\hat{c}_{\text{GE(2)}}$	0.4887	0.398	0.0569
20% censoring							
$\hat{\gamma}_{\text{AE}}$	0.2545	0.1075	0.2415	$\hat{c}_{\text{AE}}$	0.5436	0.4393	-0.2285
$\hat{\gamma}_{\text{SE}}$	0.2774	0.1243	0.2675	$\hat{c}_{\text{SE}}$	0.5378	0.4315	-0.2028
$\hat{\gamma}_{\text{LNX(-2)}}$	0.3307	0.1776	0.3223	$\hat{c}_{\text{LNX(-2)}}$	0.6103	0.6363	0.2374
$\hat{\gamma}_{\text{LNX(2)}}$	0.2389	0.0934	0.2249	$\hat{c}_{\text{LNX(2)}}$	0.667	0.6329	-0.5684
$\hat{\gamma}_{\text{GE(-2)}}$	0.3029	0.1437	0.2947	$\hat{c}_{\text{GE(-2)}}$	0.5357	0.4473	-0.1669
$\hat{\gamma}_{\text{GE(2)}}$	0.2126	0.0794	0.188	$\hat{c}_{\text{GE(2)}}$	0.5769	0.5034	-0.3308

5.1.5 Discussion

Upon inspection of the results for the GCRE model, it is immediately clear that censoring causes an overestimation of the parameter γ as well as a slight underestimation of the value of the generalisation parameter c. More specifically, while the estimates of γ and c with the highest coverage and lowest MSE and bias are generally equivalent to the true values, the censored estimates with the best properties are found at around 140% of the true γ value and around 95% of the true c value. Thus, the effect of censoring is not at all as prominent for the generalisation parameter.

Even though the performance of $\hat{\gamma}_{\text{SE}}$ and $\hat{\gamma}_{\text{AE}}$ is very similar, the latter seems to be slightly more accurate. This is consistent with findings for the previous models, especially in the presence of censoring. The MAE, MSE and bias support the superiority of $\hat{\gamma}_{\text{AE}}$ in terms of accuracy. The difference between $\hat{c}_{\text{SE}}$ and $\hat{c}_{\text{AE}}$ is even less pronounced and with censoring it would even seem that the former yields marginally lower MAE, MSE and bias than the latter.

Once again, hyperbolic-shaped monotonically decreasing curves are seen as the hyper-parameters of the asymmetric loss function estimators varies, although in this case those relating to the GE loss appear close to linear. For estimators of both γ and c, the values of a and k which yield the true parameter values does not change much for different parameter configurations, but that may only be because the ranges of parameters used were relatively narrow.

Investigation of the single case where a reduced sample size was used, it is clear that the performance was slightly worse overall, as expected.

In general it seems that the estimators of the hazard rate performed well in reproducing the shape of the true hazard rate. In some cases, censoring seems to affect the scale of the function quite drastically.

5.2 The GCRG model

5.2.1 Model characteristics

In the final part of the simulation study, one more generalisation relating to the Rayleigh distribution, derived in Section 3.2.2.3, is considered. The GCRG model has distribution function (3.10) and density function (3.12) and these are used to find the survival function

$$S(t, \alpha, \beta, c) = \left(1 + \frac{t^c}{\beta}\right)^{-\alpha}$$

and the hazard rate

$$\text{and } h(t, \alpha, \beta, c) = \frac{\alpha c t^{c-1}}{t^c + \beta}. \tag{5.8}$$

Although their form is similar to their non-generalised counterparts in Section 4.2, it can be seen that the parameter c adds an additional layer of versatility to the way in which these functions are modelled. In Figure 5.20, one can see that the hazard rate of the GCRG model can vary greatly depending on the choice of model parameters configuration.

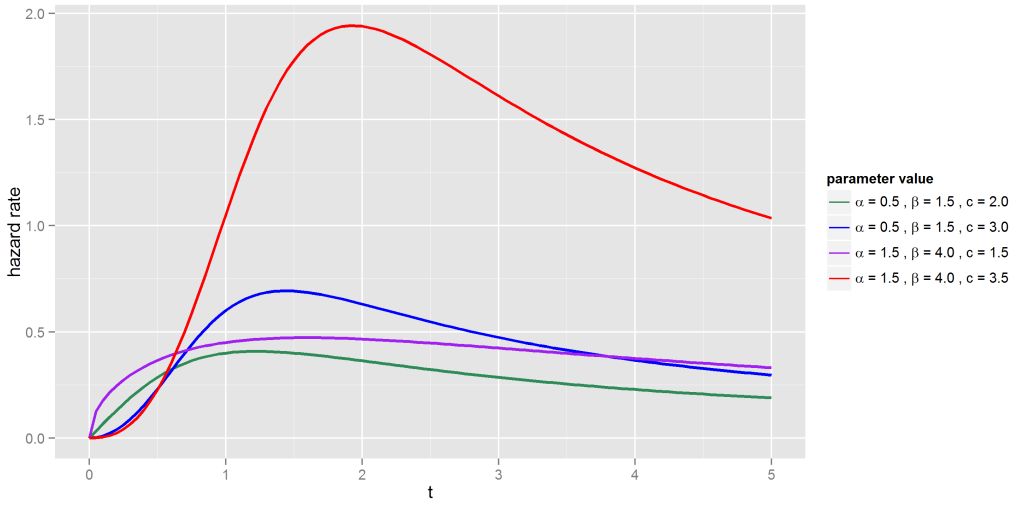


Figure 5.20: *Hazard rate of GCRG model for various values of its parameters.*

The Fisher information (2.11) now becomes a 3×3 matrix of expected values. The second-order partial derivatives of the log of the PDF (3.12) can be written from

$$l_f = \ln f(t|\alpha, \beta, c) = \ln c + \ln \alpha + \alpha \ln \beta + (c-1) \ln t - (\alpha + 1) \ln(t^c + \beta)$$

and it follows that

$$\begin{aligned}
\frac{\partial^2 l_f}{\partial \alpha^2} &= -\frac{1}{\alpha^2} \\
\frac{\partial^2 l_f}{\partial \alpha \partial \beta} &= \frac{1}{\beta} - \frac{1}{t^c + \beta} \\
\frac{\partial^2 l_f}{\partial \alpha \partial c} &= -\frac{t^c \ln t}{t^c + \beta} \\
\frac{\partial^2 l_f}{\partial \beta^2} &= -\frac{\alpha}{\beta^2} + \frac{\alpha + 1}{(t^c + \beta)^2} \\
\frac{\partial^2 l_f}{\partial \beta \partial c} &= \frac{t^c \ln t(\alpha + 1)}{(t^c + \beta)^2} \\
\frac{\partial^2 l_f}{\partial c^2} &= -\frac{1}{c^2} - \frac{(\alpha + 1)\beta t^c (\ln t)^2}{(t^c + \beta)^2}.
\end{aligned}$$

$\mathcal{I}_F$ is a square symmetric matrix, thus it can be defined by specifying the diagonal and lower triangular elements, in terms of expected values with respect to the GCRG model. These matrix elements are

$$[\mathcal{I}_F]_{11} = -\mathrm{E}\left[\frac{\partial^2 l_f}{\partial \alpha^2}\right] = \frac{1}{\alpha^2} \tag{5.9}$$

$$[\mathcal{I}_F]_{12} = [\mathcal{I}_F]_{21} = -\mathrm{E}\left[\frac{\partial^2 l_f}{\partial \alpha \partial \beta}\right] = \mathrm{E}\left[\frac{1}{T^c + \beta}\right] - \frac{1}{\beta} \tag{5.10}$$

$$[\mathcal{I}_F]_{13} = [\mathcal{I}_F]_{31} = -\mathrm{E}\left[\frac{\partial^2 l_f}{\partial \alpha \partial c}\right] = \mathrm{E}\left[\frac{T^c \ln T}{T^c + \beta}\right] \tag{5.11}$$

$$[\mathcal{I}_F]_{22} = -\mathrm{E}\left[\frac{\partial^2 l_f}{\partial \beta^2}\right] = \frac{\alpha}{\beta^2} - (\alpha + 1)\mathrm{E}\left[\frac{1}{(T^c + \beta)^2}\right] \tag{5.12}$$

$$[\mathcal{I}_F]_{23} = [\mathcal{I}_F]_{32} = -\mathrm{E}\left[\frac{\partial^2 l_f}{\partial \beta \partial c}\right] = -(\alpha + 1)\mathrm{E}\left[\frac{T^c \ln T}{(T^c + \beta)^2}\right] \tag{5.13}$$

$$[\mathcal{I}_F]_{33} = -\mathrm{E}\left[\frac{\partial^2 l_f}{\partial c^2}\right] = \frac{1}{c^2} - (\alpha + 1)\beta \mathrm{E}\left[\frac{T^c (\ln T)^2}{(T^c + \beta)^2}\right] \tag{5.14}$$

The only remaining task to acquire the information matrix is the simplification of the five expected values above. The first expectation (5.10) can be written in the form

$$\begin{aligned}
\mathrm{E}\left[\frac{1}{T^c + \beta}\right] &= \int_0^\infty \alpha c \beta^\alpha t^{c-1} (t^c + \beta)^{-\alpha - 2} dt \\
&= \alpha \int_0^\infty c \beta^{-2} t^{c-1} \left(1 + \frac{t^c}{\beta}\right)^{-\alpha - 2} dt \\
&= \alpha \beta^{-1 - \frac{1}{c}} \int_0^\infty c \left(\frac{t}{\beta^{\frac{1}{c}}}\right)^{c-1} \left[1 + \left(\frac{t}{\beta^{\frac{1}{c}}}\right)^c\right]^{-\alpha - 2} dt,
\end{aligned}$$

such that the integrand corresponds to the PDF of the Beta Prime distribution, with parameters $p \equiv c$, $q \equiv \beta^{\frac{1}{c}}$, $r \equiv 1$ and $s \equiv (\alpha + 1)$ – refer to Appendix A.2. Thus, the solution becomes

$$\mathrm{E}\left[\frac{1}{T^c + \beta}\right] = \alpha\beta^{-1-\frac{1}{c}}\beta^{\frac{1}{c}}B(1, \alpha + 1) = \frac{\alpha}{\beta(\alpha + 1)}.$$

The second expectation (5.12) with the quadratic term is solved in mostly the same way, by noting that

$$\begin{aligned}\mathrm{E}\left[\frac{1}{(T^c + \beta)^2}\right] &= \int_0^\infty \alpha c\beta^\alpha t^{c-1}(t^c + \beta)^{-\alpha-3}dt \\ &= \alpha\beta^{-2-\frac{1}{c}}\int_0^\infty c\left(\frac{t}{\beta^{\frac{1}{c}}}\right)^{c-1}\left[1 + \left(\frac{t}{\beta^{\frac{1}{c}}}\right)^c\right]^{-\alpha-3}dt\end{aligned}$$

and like the previous case, one can use the Beta Prime PDF with $s \equiv (\alpha + 2)$ to show that

$$\mathrm{E}\left[\frac{1}{(T^c + \beta)^2}\right] = \alpha\beta^{-2-\frac{1}{c}}\beta^{\frac{1}{c}}B(1, \alpha + 2) = \frac{\alpha}{\beta^2(\alpha + 2)}.$$

In the case of the GCRE model's derivation of the Fisher information matrix in Section 5.1.1, closed-form solutions could not be found analytically for all expected values. Something similar is seen in the case of the GCRG model here, whereby three of the expected values' integrals cannot be simplified. This is mainly because of the logarithmic terms in equations (5.11) to (5.14). These will be denoted by $A_1(\alpha, \beta, c)$, $A_2(\alpha, \beta, c)$ and $A_3(\alpha, \beta, c)$ from this point onwards, with the assumption that they can be approximated using the numerical technique described in Section 2.2.3.2.

$$\begin{aligned}\mathrm{E}\left[\frac{T^c \ln T}{T^c + \beta}\right] &= \int_0^\infty \alpha c\beta^\alpha \frac{t^{2c-1}\ln t}{(t^c + \beta)^{\alpha+2}}dt &&= A_1(\alpha, \beta, c) \\ \mathrm{E}\left[\frac{T^c \ln T}{(T^c + \beta)^2}\right] &= \int_0^\infty \alpha c\beta^\alpha \frac{t^{2c-1}\ln t}{(t^c + \beta)^{\alpha+3}}dt &&= A_2(\alpha, \beta, c) \\ \mathrm{E}\left[\frac{T^c(\ln T)^2}{(T^c + \beta)^2}\right] &= \int_0^\infty \alpha c\beta^\alpha \frac{t^{2c-1}(\ln t)^2}{(t^c + \beta)^{\alpha+3}}dt &&= A_3(\alpha, \beta, c)\end{aligned}$$

The calculations and simplifications above enable the final form of the Fisher information matrix

$$\mathcal{I}_F = \begin{bmatrix} \frac{1}{\alpha^2} & \frac{-1}{\beta(\alpha+1)} & A_1(\alpha,\beta,c) \\ \frac{-1}{\beta(\alpha+1)} & \frac{\alpha}{\beta^2(\alpha+2)} & -(\alpha+1)A_2(\alpha,\beta,c) \\ A_1(\alpha,\beta,c) & -(\alpha+1)A_2(\alpha,\beta,c) & \frac{1}{c^2}+(\alpha+1)\beta A_3(\alpha,\beta,c) \end{bmatrix}, \tag{5.15}$$

on which the non-informative prior derivation will be based.

5.2.2 Prior and posterior distributions

In order to do a posterior analysis of the GCRG model, a posterior distribution of the three model parameters must be found. Even though the Fisher information matrix (5.15) is written in terms of analytically insoluble integrals $A_1(\alpha,\beta,c)$, $A_2(\alpha,\beta,c)$ and $A_3(\alpha,\beta,c)$, they can be approximated.

Considering the Jeffreys prior described in Section 2.2.4.1, the multivariate prior for all three model parameters becomes

$$\pi_{\text{jeff}}(\alpha,\beta,c) \propto \sqrt{|\mathcal{I}_F|}.$$

Additionally, three-parameter reference and probability matching priors can be derived similar to those in Section 4.2.2. However, one problem with this formulation is the computational cost associated with three numerical integrations that needs to be processed each of the thousands of times the calculation of $A_1(\alpha,\beta,c)$, $A_2(\alpha,\beta,c)$ and $A_3(\alpha,\beta,c)$ are required in the simulation runs. This computational problem can be attenuated by considering conditional partitioning of the prior distribution. Using standard rules of probability theory, one can say

$$\pi(\alpha,\beta,c) = \pi(\alpha,\beta)\pi(c|\alpha,\beta).$$

Thus, the conditional prior of the generalisation parameter c, given α and β, can be used in a product with the prior of α and β to form a prior distribution for all three parameters, reducing the amount of numerical approximations threefold. The prior $\pi(\alpha,\beta)$ corresponds to the upper left four elements of the matrix (5.15). Since this sub-matrix is equivalent to the Fisher information matrix for the non-generalised case (4.9),

the priors derived for the CRG model can be reused here, such that

$$\begin{aligned}\pi_{\text{jeff}}(\alpha,\beta) &\propto \frac{1}{\beta(\alpha+1)\sqrt{\alpha(\alpha+2)}}\\ \pi_{\text{ref}}(\alpha,\beta) &\propto \frac{1}{\alpha\beta}\\ \pi_{\text{PM}}(\alpha,\beta) &\propto \frac{1}{\alpha\beta(\alpha+1)}\end{aligned}$$

from equations (4.11), (4.12) and (4.15) in Section 4.2.2.

The prior of c, $\pi(c)$, is derived non-informatively, for which a natural choice is the Jeffreys prior. Thus, the bottom right diagonal element of (5.15) is used, such that

$$\pi_{\text{jeff}}(c|\alpha,\beta) \propto \sqrt{\frac{1}{c^2}+(\alpha+1)c\alpha\beta^{\alpha+1}A_3},$$

and due to the fact that this is a single-dimensional prior, one can define

$$\pi_{\text{jeff}}(c|\alpha,\beta) = \pi_{\text{ref}}(c|\alpha,\beta) = \pi_{\text{PM}}(c|\alpha,\beta).$$

Consequently, from the equations above, the proportional forms of the three prior distributions for the GCRG model follow as

$$\begin{aligned}\pi_{\text{jeff}}(\alpha,\beta,c) &= \pi_{\text{jeff}}(\alpha,\beta)\pi_{\text{jeff}}(c|\alpha,\beta)\\ \pi_{\text{ref}}(\alpha,\beta,c) &= \pi_{\text{ref}}(\alpha,\beta)\pi_{\text{ref}}(c|\alpha,\beta)\\ \pi_{\text{PM}}(\alpha,\beta,c) &= \pi_{\text{PM}}(\alpha,\beta)\pi_{\text{PM}}(c|\alpha,\beta).\end{aligned} \tag{5.16}$$

In addition to the parameters' prior distribution, the likelihood function is required. To this end, consider a sample of n survival times $\mathbf{t} = (t_1, t_2, \ldots, t_n)$, ordered such that the first d are non-censored and the remaining $(n-d)$ right censored. The likelihood then becomes

$$\begin{aligned}\mathcal{L}(\alpha,\beta,c|\mathbf{t}) &\propto \prod_{i=1}^{d} f(t_i|\alpha,\beta,c)\prod_{j=d+1}^{n} S(t_j,\alpha,\beta,c)\\ &\propto \prod_{i=1}^{d}\frac{c\alpha\beta^{\alpha}t_i^{c-1}}{(t_i^c+\beta)^{\alpha+1}}\prod_{j=d+1}^{n}\left(1+\frac{t_j^c}{\beta}\right)^{-\alpha}\\ &\propto (c\alpha)^d\prod_{i=1}^{d}\frac{t_i^{c-1}}{t_i^c+\beta}\prod_{j=1}^{n}\left(1+\frac{t_j^c}{\beta}\right)^{-\alpha}.\end{aligned}$$

For computational reasons, the products in the last step are written as logarithms, resulting in

$$\begin{aligned}\mathcal{L}(\alpha,\beta,c|\mathbf{t}) &\propto (c\alpha)^d e^{W_1(\beta,c)-\alpha W_2(\beta,c)}\\ \text{where } W_1(\beta,c) &= \sum_{i=1}^{d}\left[(c-1)\ln t_i - \ln(t_i^c+\beta)\right]\\ \text{and } W_2(\beta,c) &= \sum_{i=1}^{n}\ln\left(1+\frac{t_i^c}{\beta}\right).\end{aligned}$$

The proportional forms of the posterior distributions now follow from the products of the likelihood function above with each of the priors (5.16) respectively.

5.2.3 Bayesian estimators of the parameters

The posterior distribution of the parameters is required to formally define the Bayesian estimators of interest, along with a preselected loss function. For each of the loss functions under consideration, the definitions that follow are in terms of an arbitrary posterior distribution of the model parameters. In the simulation study, each of these estimators are computed using all three of the posterior distribution corresponding to the Jeffreys, reference and PM priors, discussed in the previous section.

Similar to previous cases, under the symmetric AE and SE loss functions, the estimators become the posterior expected values and medians

$$\begin{aligned}\left(\hat{\alpha}_{\text{AE}},\ \hat{\beta}_{\text{AE}},\ \hat{c}_{\text{AE}}\right) &= \text{median}_{\alpha,\beta,c|\mathbf{t}}[(\alpha,\beta,c)]\\ \left(\hat{\alpha}_{\text{SE}},\ \hat{\beta}_{\text{SE}},\ \hat{c}_{\text{SE}}\right) &= \text{E}_{\alpha,\beta,c|\mathbf{t}}[(\alpha,\beta,c)].\end{aligned}$$

Using the asymmetric LINEX loss function with its parameter a, and the GE loss function with its parameter k, the Bayesian estimators are given by

$$\begin{aligned}\left(\hat{\alpha}_{\text{LNX(a)}},\ \hat{\beta}_{\text{LNX(a)}},\ \hat{c}_{\text{LNX(a)}}\right) &= -\frac{1}{a}\ln \text{E}_{\alpha,\beta,c|\mathbf{t}}\left[e^{-a(\alpha,\beta,c)}\right]\\ \left(\hat{\alpha}_{\text{GE(k)}},\ \hat{\beta}_{\text{GE(k)}},\ \hat{c}_{\text{GE(k)}}\right) &= \left(\text{E}_{\alpha,\beta,c|\mathbf{t}}\left[(\alpha,\beta,c)^{-k}\right]\right)^{-\frac{1}{k}}.\end{aligned}$$

The derivation of Bayesian estimators for functions of the parameters of interest yield analogous results to those above. Only those of the hazard rate are shown here and

investigated in the simulation study. Their forms are given by

$$\begin{aligned}
\hat{h}\mathrm{SE}(t,\alpha,\beta,c) &= \mathrm{E}_{\alpha,\beta,c|\mathbf{t}}[h(t,\alpha,\beta,c)] \\
\hat{h}_{\mathrm{AE}}(t,\alpha,\beta,c) &= \mathrm{median}_{\alpha,\beta,c|\mathbf{t}}[h(t,\alpha,\beta,c)] \\
\hat{h}_{\mathrm{LNX(a)}}(t,\alpha,\beta,c) &= -\frac{1}{a}\ln \mathrm{E}_{\alpha,\beta,c|\mathbf{t}}\left[e^{-ah(t,\alpha,\beta,c)}\right] \\
\hat{h}_{\mathrm{GE(k)}}(t,\alpha,\beta,c) &= \left(\mathrm{E}_{\alpha,\beta,c|\mathbf{t}}\left[h(t,\alpha,\beta,c)^{-k}\right]\right)^{-\frac{1}{k}}.
\end{aligned}$$

5.2.4 Simulation results

The simulation study was performed in accordance with the procedures set out in Section 3.3.2 to assess properties of different Bayesian estimators and prior distributions derived under the GCRG model. This process was carried out for four different configurations of parameter values

$$\begin{aligned}
&\alpha = 0.5\ , \qquad \beta = 1.5\ , \qquad c = 2.0 \\
&\alpha = 0.5\ , \qquad \beta = 1.5\ , \qquad c = 3.0 \\
&\alpha = 1.5\ , \qquad \beta = 4.0\ , \qquad c = 1.5 \\
&\alpha = 1.5\ , \qquad \beta = 4.0\ , \qquad c = 3.5
\end{aligned}$$

and two levels of censoring: $\delta = \{1, 0.8\}$. Note that the parameter configurations are split into two sets where α and β were kept fixed, while c was varied. It was chosen in this way so that the estimators could be compared across changing values of the generalisation parameter, the main addition in this chapter. In all cases, a sample size of 50 was used, except for the parameter configuration ($\alpha = 0.5$, $\beta = 1.5$, $c = 3.0$), for which both $n = 30$ and $n = 50$ were considered.

The results of the simulations are presented in a similar fashion to the previous models' results, mainly consisting of plots for which descriptions can be found in Section 4.1.4. This includes plots of Bayesian point estimates (Figures 5.21, 5.26, 5.36 and 5.41), plots of Bayesian estimates of the hazard rate (Figures 5.25, 5.30, 5.40 and 5.45) and plots of coverages and MSE vs bias (Figures 5.22 to 5.24, 5.27 to 5.29, 5.37 to 5.39 and 5.42 to 5.44). Additionally, Tables 5.7, 5.8, 5.10 and 5.11 summarise the measures of accuracy (MSE, MAE and bias). A similar set of results for the case of $n = 30$ is shown in Figures 5.31, 5.35, 5.32 to 5.34 and Table 5.9.

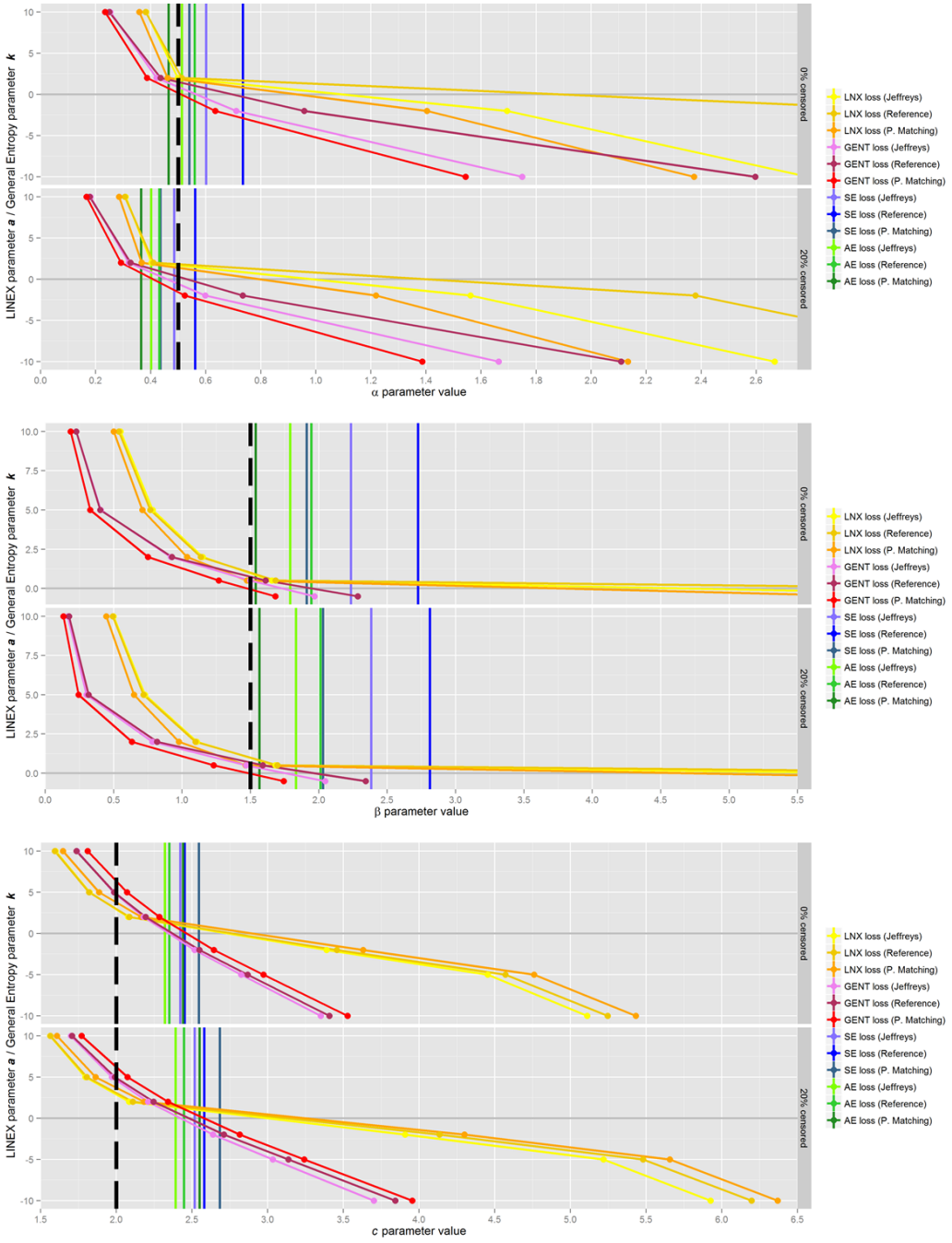


Figure 5.21: *Bayesian estimates plot for GCRG model, with* $\alpha = 0.5$ *(top),* $\beta = 1.5$ *(middle) and* $c = 2$ *(bottom), and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours).*

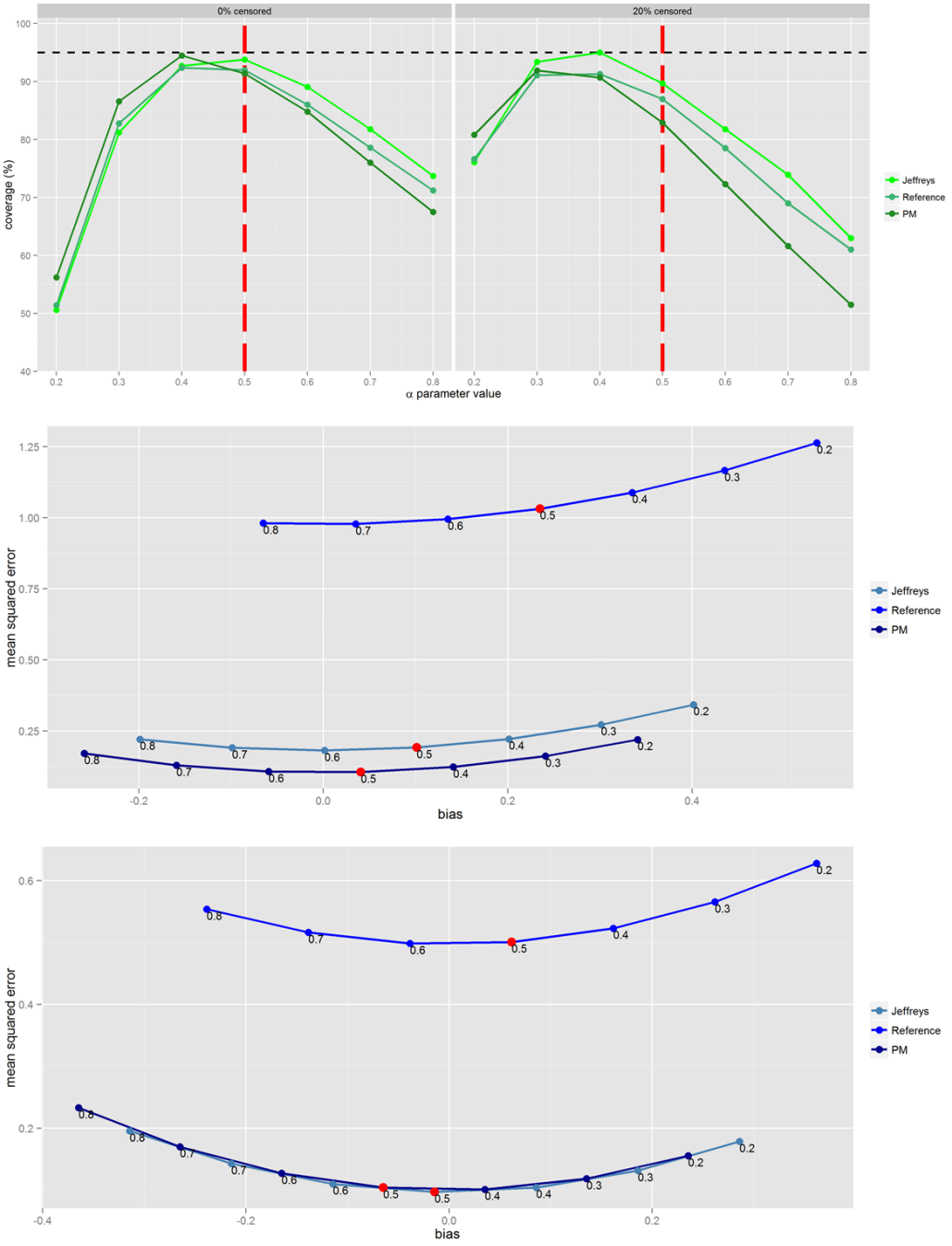


Figure 5.22: *Coverage plots (top) as well as MSE vs bias plots (bottom two) for GCRG model, with* $\alpha = 0.5$ *(*$\beta = 1.5$ *and* $c = 2$*), for all prior distributions and* $\delta = 1$ *(no censoring) and* $\delta = 0.8$ *(20% censored values).*

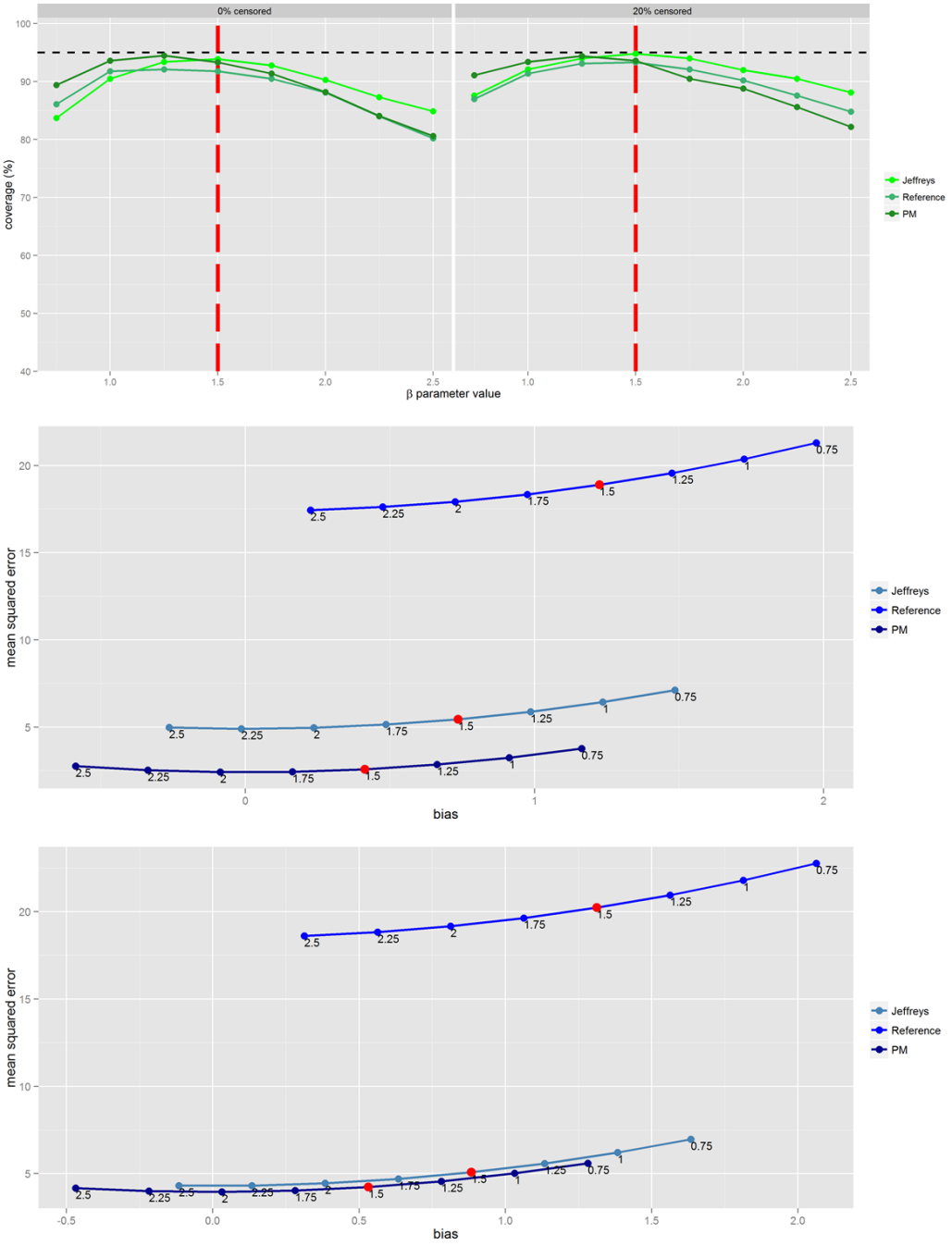


Figure 5.23: *Coverage plots (top) as well as MSE vs bias plots (bottom two) for GCRG model, with* $\beta = 1.5$ *(*$\alpha = 0.5$ *and* $c = 2$*), for all prior distributions and* $\delta = 1$ *(no censoring) and* $\delta = 0.8$ *(20% censored values).*

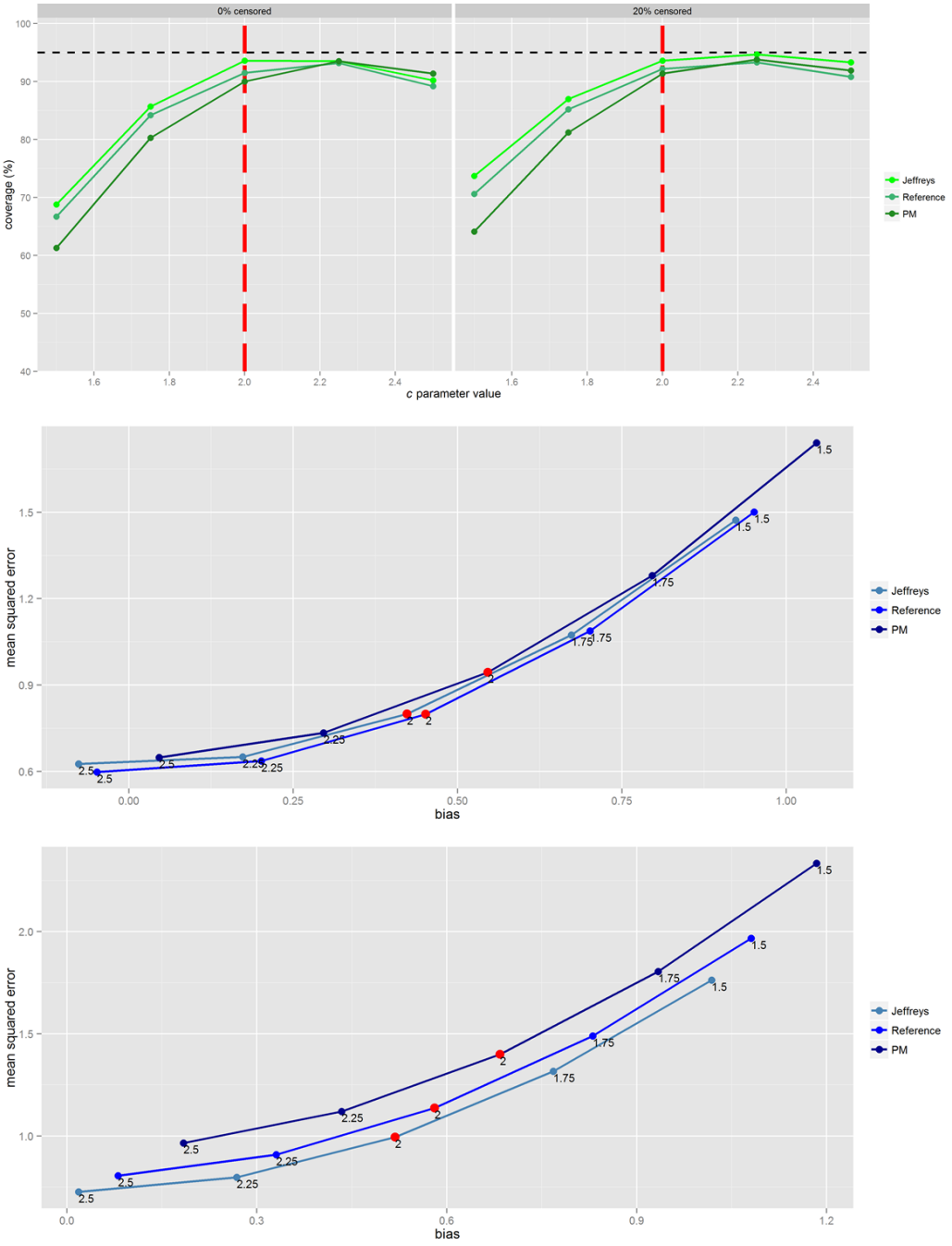


Figure 5.24: *Coverage plots (top) as well as MSE vs bias plots (bottom two) for GCRG model, with* $c = 2$ *(*$\alpha = 0.5$ *and* $\beta = 1.5$*), for all prior distributions and* $\delta = 1$ *(no censoring) and* $\delta = 0.8$ *(20% censored values).*

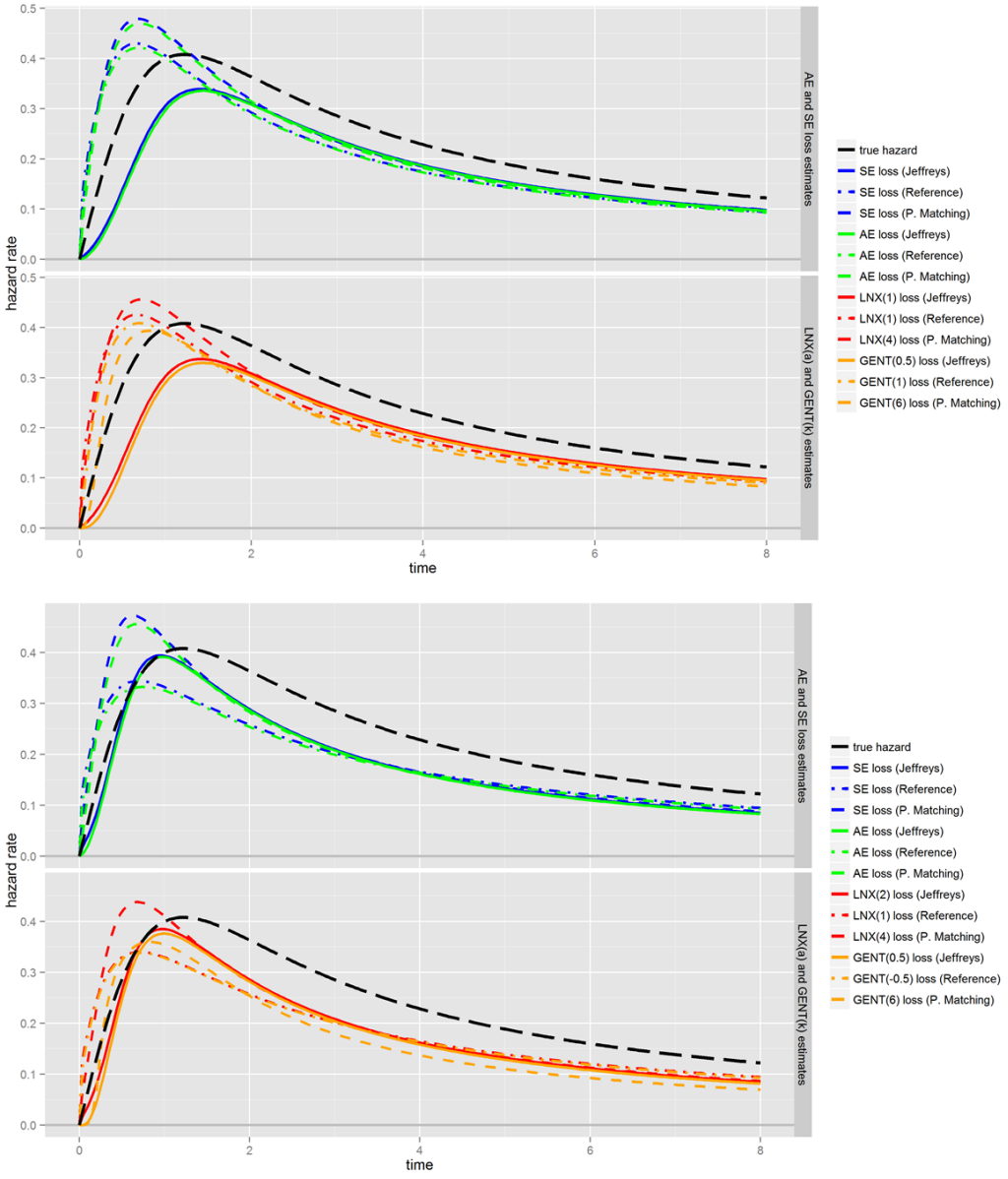


Figure 5.25: *Plots of Bayesian estimates of the hazard function for the GCRG model with* $(\alpha, \beta, c) = (0.5, 1.5, 2)$ *and two levels of censoring (no censoring, top, and 20% censoring, bottom), derived using three different priors and four different loss functions.*

Table 5.7: *The MAE, MSE and bias of estimators for the GCRG model, with parameters* $(\alpha, \beta, c) = (0.5, 1.5, 3)$.

prior	estimator	MAE	MSE	bias	estimator	MAE	MSE	bias	estimator	MAE	MSE	bias
					0% censoring							
Jeffreys	$\hat{\alpha}_{\mathrm{AE}}$	0.19	0.081	0.013	$\hat{\beta}_{\mathrm{AE}}$	0.956	2.428	0.291	$\hat{c}_{\mathrm{AE}}$	0.512	0.633	0.322
	$\hat{\alpha}_{\mathrm{SE}}$	0.237	0.192	0.101	$\hat{\beta}_{\mathrm{SE}}$	1.212	5.445	0.736	$\hat{c}_{\mathrm{SE}}$	0.577	0.799	0.423
	$\hat{\alpha}_{\mathrm{LNX}(2)}$	0.165	0.048	-0.001	$\hat{\beta}_{\mathrm{LNX}(2)}$	0.613	0.514	-0.349	$\hat{c}_{\mathrm{LNX}(2)}$	0.349	0.231	0.078
	$\hat{\alpha}_{\mathrm{GE}(2)}$	0.177	0.048	-0.078	$\hat{\beta}_{\mathrm{GE}(2)}$	0.965	1.321	-0.582	$\hat{c}_{\mathrm{GE}(2)}$	0.429	0.425	0.179
Reference	$\hat{\alpha}_{\mathrm{AE}}$	0.251	0.278	0.06	$\hat{\beta}_{\mathrm{AE}}$	1.171	5.655	0.446	$\hat{c}_{\mathrm{AE}}$	0.564	0.646	0.35
	$\hat{\alpha}_{\mathrm{SE}}$	0.383	1.032	0.235	$\hat{\beta}_{\mathrm{SE}}$	1.753	18.896	1.225	$\hat{c}_{\mathrm{SE}}$	0.625	0.799	0.451
	$\hat{\alpha}_{\mathrm{LNX}(2)}$	0.191	0.079	0.013	$\hat{\beta}_{\mathrm{LNX}(2)}$	0.641	0.589	-0.363	$\hat{c}_{\mathrm{LNX}(2)}$	0.377	0.244	0.086
	$\hat{\alpha}_{\mathrm{GE}(2)}$	0.204	0.086	-0.064	$\hat{\beta}_{\mathrm{GE}(2)}$	1.04	1.75	-0.574	$\hat{c}_{\mathrm{GE}(2)}$	0.466	0.419	0.194
PM	$\hat{\alpha}_{\mathrm{AE}}$	0.178	0.055	-0.034	$\hat{\beta}_{\mathrm{AE}}$	0.852	1.408	0.039	$\hat{c}_{\mathrm{AE}}$	0.598	0.759	0.442
	$\hat{\alpha}_{\mathrm{SE}}$	0.209	0.106	0.041	$\hat{\beta}_{\mathrm{SE}}$	1.006	2.58	0.413	$\hat{c}_{\mathrm{SE}}$	0.668	0.945	0.546
	$\hat{\alpha}_{\mathrm{LNX}(2)}$	0.16	0.041	-0.039	$\hat{\beta}_{\mathrm{LNX}(2)}$	0.639	0.546	-0.464	$\hat{c}_{\mathrm{LNX}(2)}$	0.396	0.285	0.164
	$\hat{\alpha}_{\mathrm{GE}(2)}$	0.18	0.045	-0.114	$\hat{\beta}_{\mathrm{GE}(2)}$	0.989	1.232	-0.75	$\hat{c}_{\mathrm{GE}(2)}$	0.498	0.505	0.284
					20% censoring							
Jeffreys	$\hat{\alpha}_{\mathrm{AE}}$	0.19	0.055	-0.098	$\hat{\beta}_{\mathrm{AE}}$	1.016	2.39	0.334	$\hat{c}_{\mathrm{AE}}$	0.579	0.765	0.392
	$\hat{\alpha}_{\mathrm{SE}}$	0.211	0.097	-0.014	$\hat{\beta}_{\mathrm{SE}}$	1.333	5.078	0.884	$\hat{c}_{\mathrm{SE}}$	0.658	0.995	0.518
	$\hat{\alpha}_{\mathrm{LNX}(2)}$	0.171	0.042	-0.097	$\hat{\beta}_{\mathrm{LNX}(2)}$	0.619	0.525	-0.391	$\hat{c}_{\mathrm{LNX}(2)}$	0.364	0.242	0.084
	$\hat{\alpha}_{\mathrm{GE}(2)}$	0.212	0.057	-0.18	$\hat{\beta}_{\mathrm{GE}(2)}$	1.041	1.416	-0.717	$\hat{c}_{\mathrm{GE}(2)}$	0.465	0.468	0.209
Reference	$\hat{\alpha}_{\mathrm{AE}}$	0.241	0.176	-0.069	$\hat{\beta}_{\mathrm{AE}}$	1.273	7.892	0.512	$\hat{c}_{\mathrm{AE}}$	0.641	0.868	0.447
	$\hat{\alpha}_{\mathrm{SE}}$	0.307	0.501	0.062	$\hat{\beta}_{\mathrm{SE}}$	1.822	20.235	1.313	$\hat{c}_{\mathrm{SE}}$	0.731	1.136	0.581
	$\hat{\alpha}_{\mathrm{LNX}(2)}$	0.198	0.061	-0.09	$\hat{\beta}_{\mathrm{LNX}(2)}$	0.67	0.633	-0.403	$\hat{c}_{\mathrm{LNX}(2)}$	0.39	0.267	0.109
	$\hat{\alpha}_{\mathrm{GE}(2)}$	0.235	0.075	-0.174	$\hat{\beta}_{\mathrm{GE}(2)}$	1.141	2.118	-0.683	$\hat{c}_{\mathrm{GE}(2)}$	0.504	0.514	0.247
PM	$\hat{\alpha}_{\mathrm{AE}}$	0.214	0.068	-0.135	$\hat{\beta}_{\mathrm{AE}}$	0.971	2.115	0.065	$\hat{c}_{\mathrm{AE}}$	0.71	1.105	0.551
	$\hat{\alpha}_{\mathrm{SE}}$	0.224	0.104	-0.065	$\hat{\beta}_{\mathrm{SE}}$	1.17	4.236	0.531	$\hat{c}_{\mathrm{SE}}$	0.806	1.4	0.684
	$\hat{\alpha}_{\mathrm{LNX}(2)}$	0.196	0.052	-0.133	$\hat{\beta}_{\mathrm{LNX}(2)}$	0.696	0.622	-0.523	$\hat{c}_{\mathrm{LNX}(2)}$	0.418	0.317	0.183
	$\hat{\alpha}_{\mathrm{GE}(2)}$	0.239	0.069	-0.208	$\hat{\beta}_{\mathrm{GE}(2)}$	1.108	1.489	-0.866	$\hat{c}_{\mathrm{GE}(2)}$	0.557	0.649	0.341

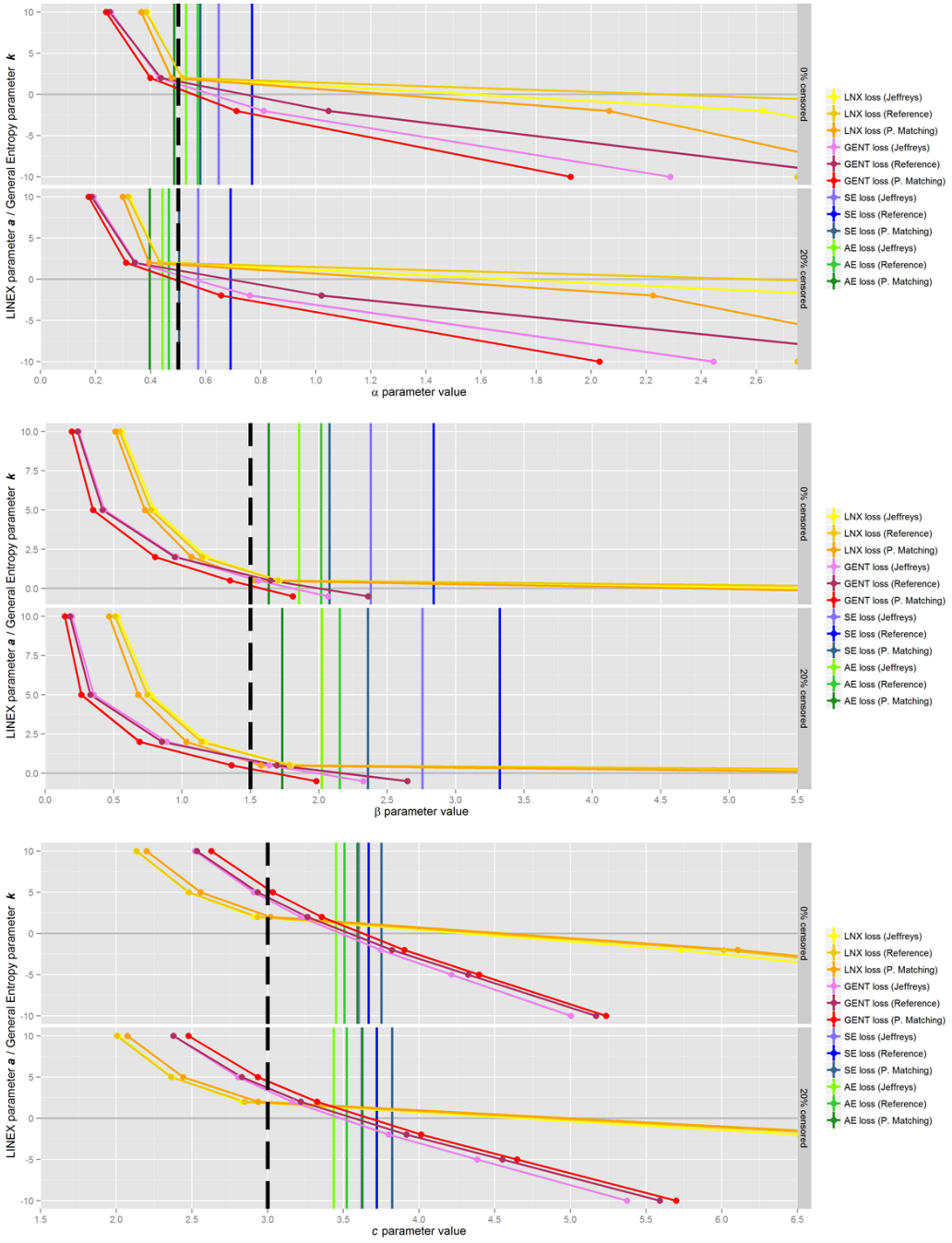


Figure 5.26: *Bayesian estimates plot for GCRG model, with* $\alpha = 0.5$ *(top),* $\beta = 1.5$ *(middle) and* $c = 3$ *(bottom), and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours).*

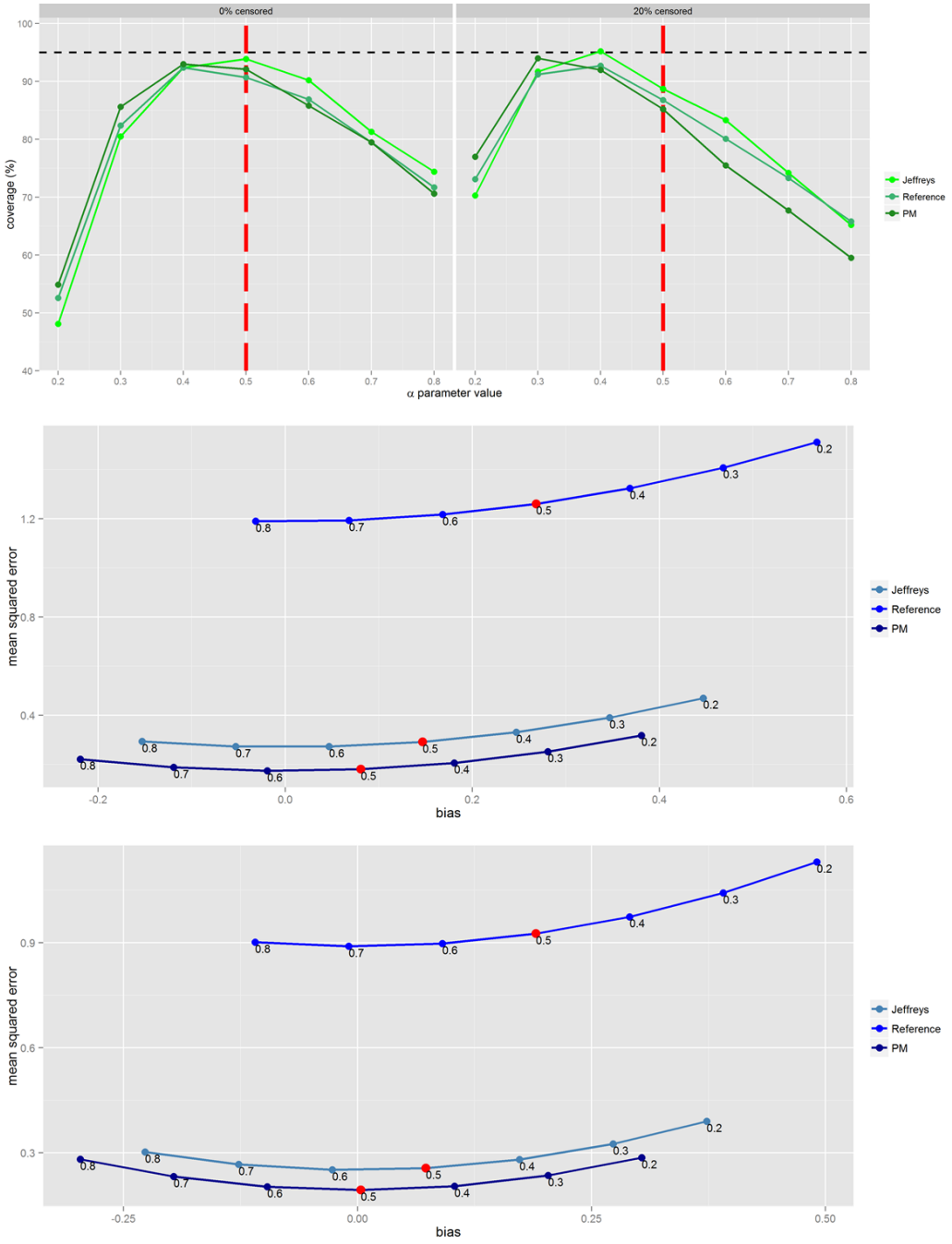


Figure 5.27: *Coverage plots (top) as well as MSE vs bias plots (bottom two) for GCRG model, with* $\alpha = 0.5$ *(*$\beta = 1.5$ *and* $c = 3$*), for all prior distributions and* $\delta = 1$ *(no censoring) and* $\delta = 0.8$ *(20% censored values).*

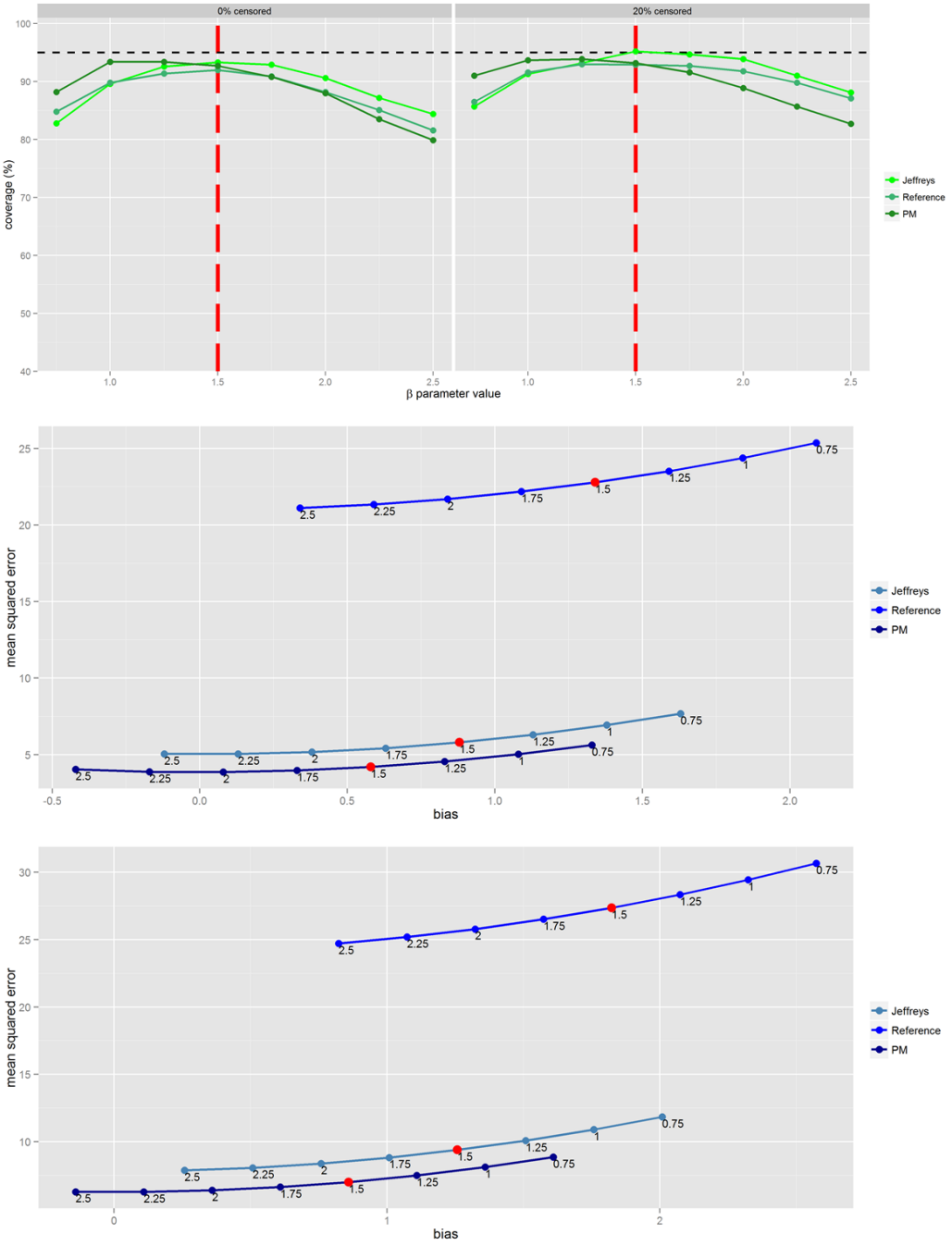


Figure 5.28: *Coverage plots (top) as well as MSE vs bias plots (bottom two) for GCRG model, with* $\beta = 1.5$ *(*$\alpha = 0.5$ *and* $c = 3$*), for all prior distributions and* $\delta = 1$ *(no censoring) and* $\delta = 0.8$ *(20% censored values).*

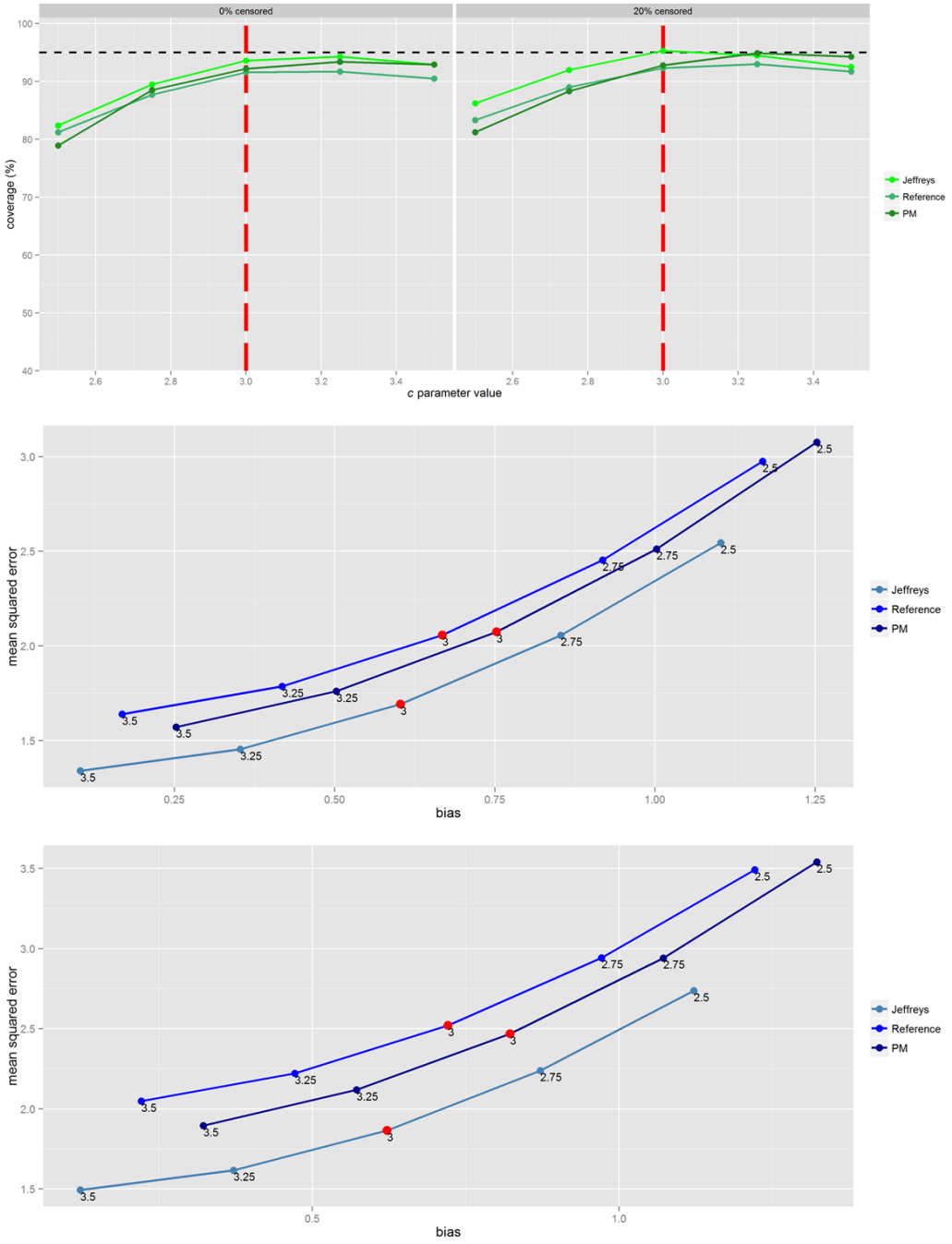


Figure 5.29: *Coverage plots (top) as well as MSE vs bias plots (bottom two) for GCRG model, with* $c = 3$ *(*$\alpha = 0.5$ *and* $\beta = 1.5$*), for all prior distributions and* $\delta = 1$ *(no censoring) and* $\delta = 0.8$ *(20% censored values).*

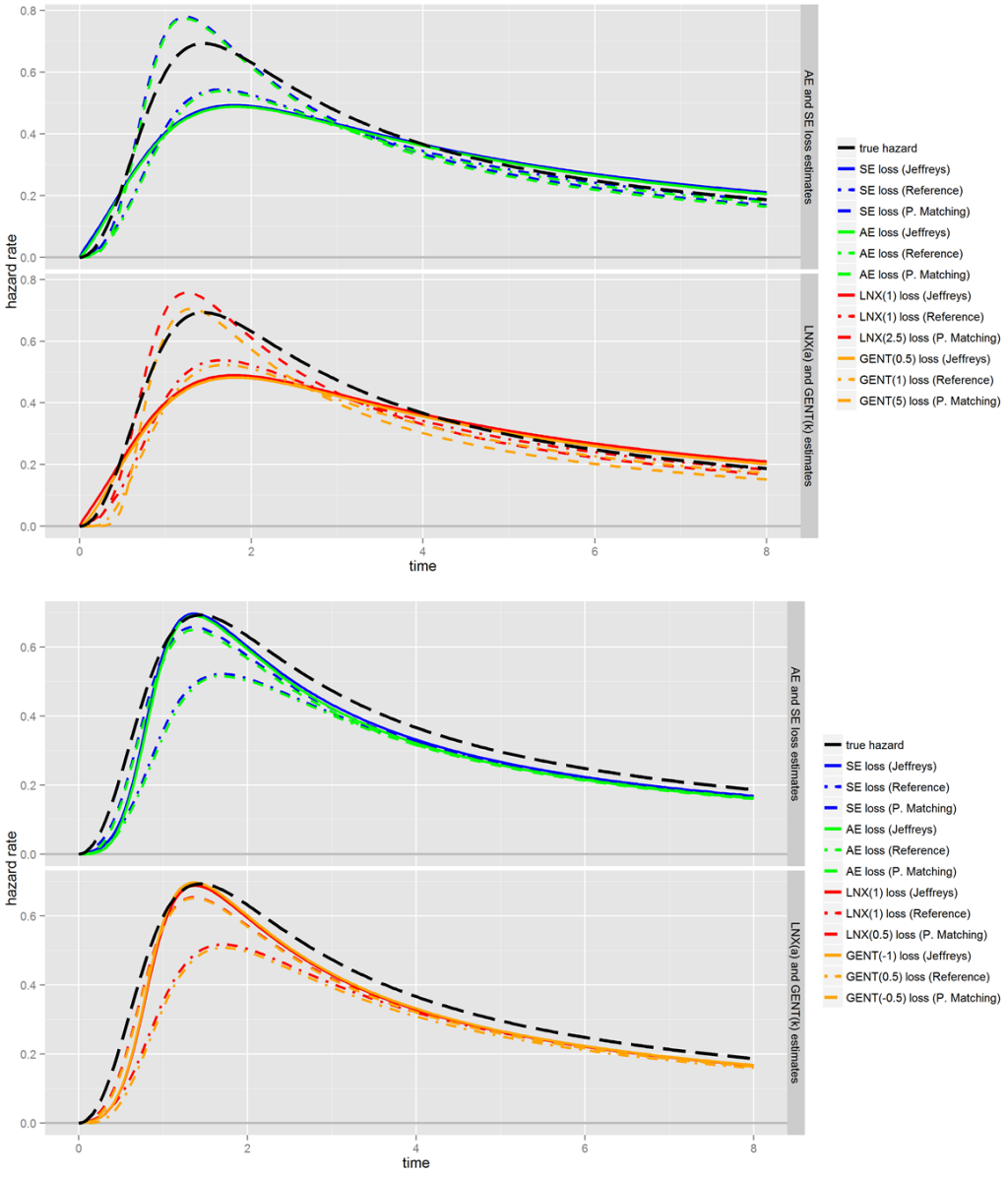


Figure 5.30: *Plots of Bayesian estimates of the hazard function for the GCRG model with* $(\alpha, \beta, c) = (0.5, 1.5, 3)$ *and two levels of censoring (no censoring, top, and 20% censoring, bottom), derived using three different priors and four different loss functions.*

Table 5.8: *The MAE, MSE and bias of estimators for the GCRG model, with parameters* $(\alpha, \beta, c) = (0.5, 1.5, 2)$.

prior	estimator	MAE	MSE	bias	estimator	MAE	MSE	bias	estimator	MAE	MSE	bias
0% censoring												
Jeffreys	$\hat{\alpha}_{\text{AE}}$	0.202	0.088	0.029	$\hat{\beta}_{\text{AE}}$	1.039	2.392	0.356	$\hat{c}_{\text{AE}}$	0.801	1.353	0.452
	$\hat{\alpha}_{\text{SE}}$	0.279	0.291	0.147	$\hat{\beta}_{\text{SE}}$	1.371	5.799	0.88	$\hat{c}_{\text{SE}}$	0.887	1.692	0.603
	$\hat{\alpha}_{\text{LNX(2)}}$	0.173	0.052	0.012	$\hat{\beta}_{\text{LNX(2)}}$	0.631	0.547	-0.323	$\hat{c}_{\text{LNX(2)}}$	0.509	0.406	-0.09
	$\hat{\alpha}_{\text{GE(2)}}$	0.179	0.049	-0.069	$\hat{\beta}_{\text{GE(2)}}$	0.986	1.325	-0.541	$\hat{c}_{\text{GE(2)}}$	0.676	0.881	0.219
Reference	$\hat{\alpha}_{\text{AE}}$	0.258	0.41	0.071	$\hat{\beta}_{\text{AE}}$	1.229	8.352	0.517	$\hat{c}_{\text{AE}}$	0.851	1.643	0.508
	$\hat{\alpha}_{\text{SE}}$	0.41	1.261	0.268	$\hat{\beta}_{\text{SE}}$	1.845	22.787	1.34	$\hat{c}_{\text{SE}}$	0.954	2.057	0.668
	$\hat{\alpha}_{\text{LNX(2)}}$	0.189	0.081	0.014	$\hat{\beta}_{\text{LNX(2)}}$	0.649	0.627	-0.355	$\hat{c}_{\text{LNX(2)}}$	0.525	0.474	-0.068
	$\hat{\alpha}_{\text{GE(2)}}$	0.205	0.1	-0.064	$\hat{\beta}_{\text{GE(2)}}$	1.071	2.142	-0.554	$\hat{c}_{\text{GE(2)}}$	0.718	1.103	0.263
PM	$\hat{\alpha}_{\text{AE}}$	0.187	0.076	-0.014	$\hat{\beta}_{\text{AE}}$	0.917	1.929	0.134	$\hat{c}_{\text{AE}}$	0.85	1.655	0.593
	$\hat{\alpha}_{\text{SE}}$	0.237	0.18	0.081	$\hat{\beta}_{\text{SE}}$	1.155	4.192	0.579	$\hat{c}_{\text{SE}}$	0.955	2.074	0.752
	$\hat{\alpha}_{\text{LNX(2)}}$	0.163	0.047	-0.025	$\hat{\beta}_{\text{LNX(2)}}$	0.638	0.553	-0.433	$\hat{c}_{\text{LNX(2)}}$	0.503	0.438	0.018
	$\hat{\alpha}_{\text{GE(2)}}$	0.183	0.05	-0.102	$\hat{\beta}_{\text{GE(2)}}$	0.999	1.322	-0.696	$\hat{c}_{\text{GE(2)}}$	0.711	1.11	0.357
20% censoring												
Jeffreys	$\hat{\alpha}_{\text{AE}}$	0.212	0.088	-0.057	$\hat{\beta}_{\text{AE}}$	1.149	3.433	0.522	$\hat{c}_{\text{AE}}$	0.834	1.425	0.437
	$\hat{\alpha}_{\text{SE}}$	0.281	0.256	0.073	$\hat{\beta}_{\text{SE}}$	1.678	9.385	1.258	$\hat{c}_{\text{SE}}$	0.942	1.865	0.622
	$\hat{\alpha}_{\text{LNX(2)}}$	0.183	0.051	-0.067	$\hat{\beta}_{\text{LNX(2)}}$	0.612	0.525	-0.327	$\hat{c}_{\text{LNX(2)}}$	0.539	0.43	-0.171
	$\hat{\alpha}_{\text{GE(2)}}$	0.211	0.061	-0.155	$\hat{\beta}_{\text{GE(2)}}$	1.033	1.485	-0.611	$\hat{c}_{\text{GE(2)}}$	0.705	0.908	0.163
Reference	$\hat{\alpha}_{\text{AE}}$	0.248	0.2	-0.034	$\hat{\beta}_{\text{AE}}$	1.308	6.985	0.654	$\hat{c}_{\text{AE}}$	0.936	1.935	0.523
	$\hat{\alpha}_{\text{SE}}$	0.408	0.926	0.191	$\hat{\beta}_{\text{SE}}$	2.253	27.363	1.823	$\hat{c}_{\text{SE}}$	1.055	2.52	0.721
	$\hat{\alpha}_{\text{LNX(2)}}$	0.194	0.062	-0.067	$\hat{\beta}_{\text{LNX(2)}}$	0.625	0.553	-0.357	$\hat{c}_{\text{LNX(2)}}$	0.583	0.505	-0.152
	$\hat{\alpha}_{\text{GE(2)}}$	0.227	0.072	-0.158	$\hat{\beta}_{\text{GE(2)}}$	1.08	1.706	-0.646	$\hat{c}_{\text{GE(2)}}$	0.787	1.169	0.218
PM	$\hat{\alpha}_{\text{AE}}$	0.211	0.079	-0.103	$\hat{\beta}_{\text{AE}}$	1.033	2.756	0.234	$\hat{c}_{\text{AE}}$	0.949	1.904	0.625
	$\hat{\alpha}_{\text{SE}}$	0.256	0.194	0.004	$\hat{\beta}_{\text{SE}}$	1.415	6.993	0.859	$\hat{c}_{\text{SE}}$	1.077	2.468	0.822
	$\hat{\alpha}_{\text{LNX(2)}}$	0.187	0.052	-0.106	$\hat{\beta}_{\text{LNX(2)}}$	0.657	0.589	-0.468	$\hat{c}_{\text{LNX(2)}}$	0.544	0.454	-0.063
	$\hat{\alpha}_{\text{GE(2)}}$	0.227	0.067	-0.189	$\hat{\beta}_{\text{GE(2)}}$	1.092	1.571	-0.81	$\hat{c}_{\text{GE(2)}}$	0.773	1.175	0.326

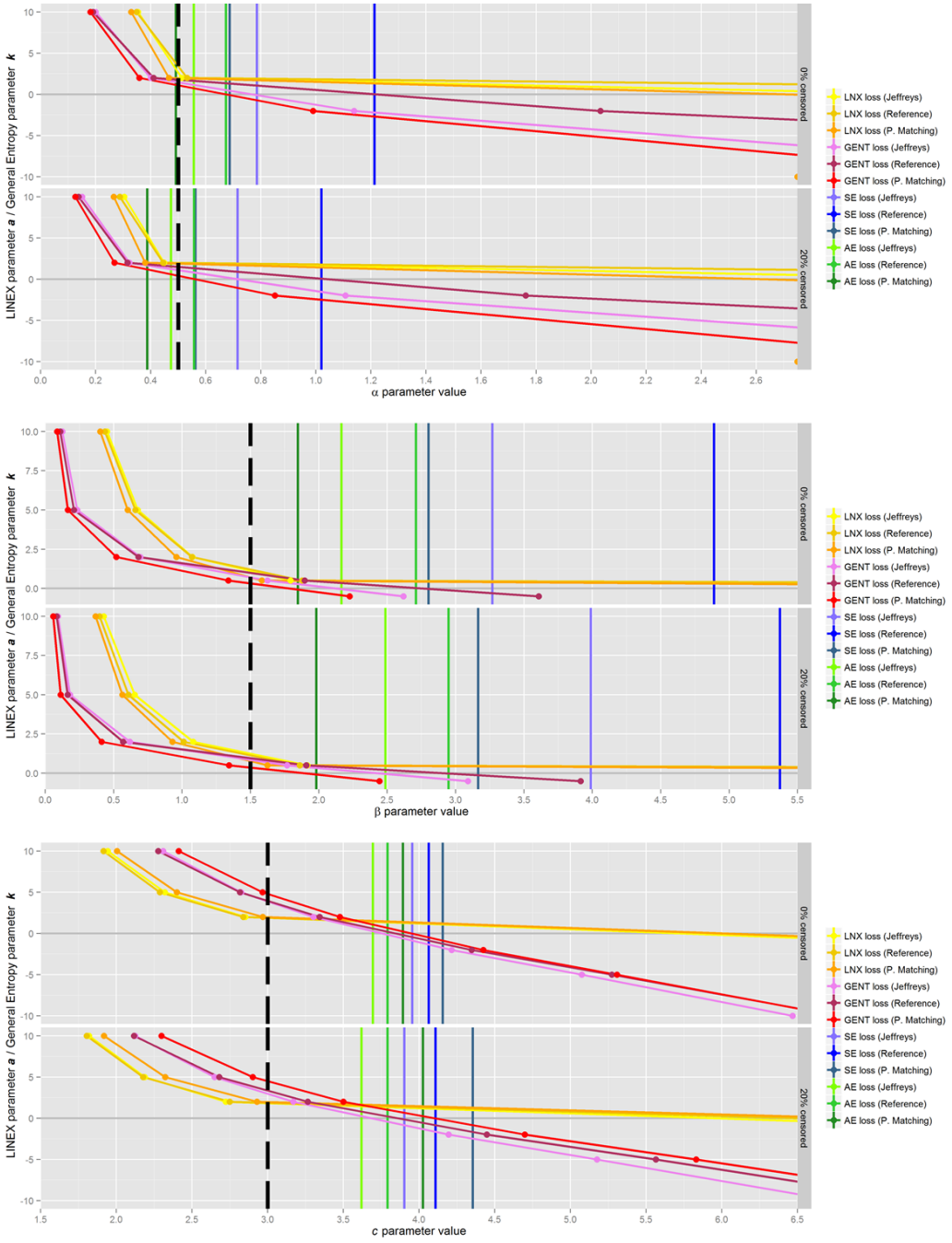


Figure 5.31: *Bayesian estimates plot for GCRG model, with* $\alpha = 0.5$ *(top),* $\beta = 1.5$ *(middle) and* $c = 3$ *(bottom), and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours), and with* $n = 30$.

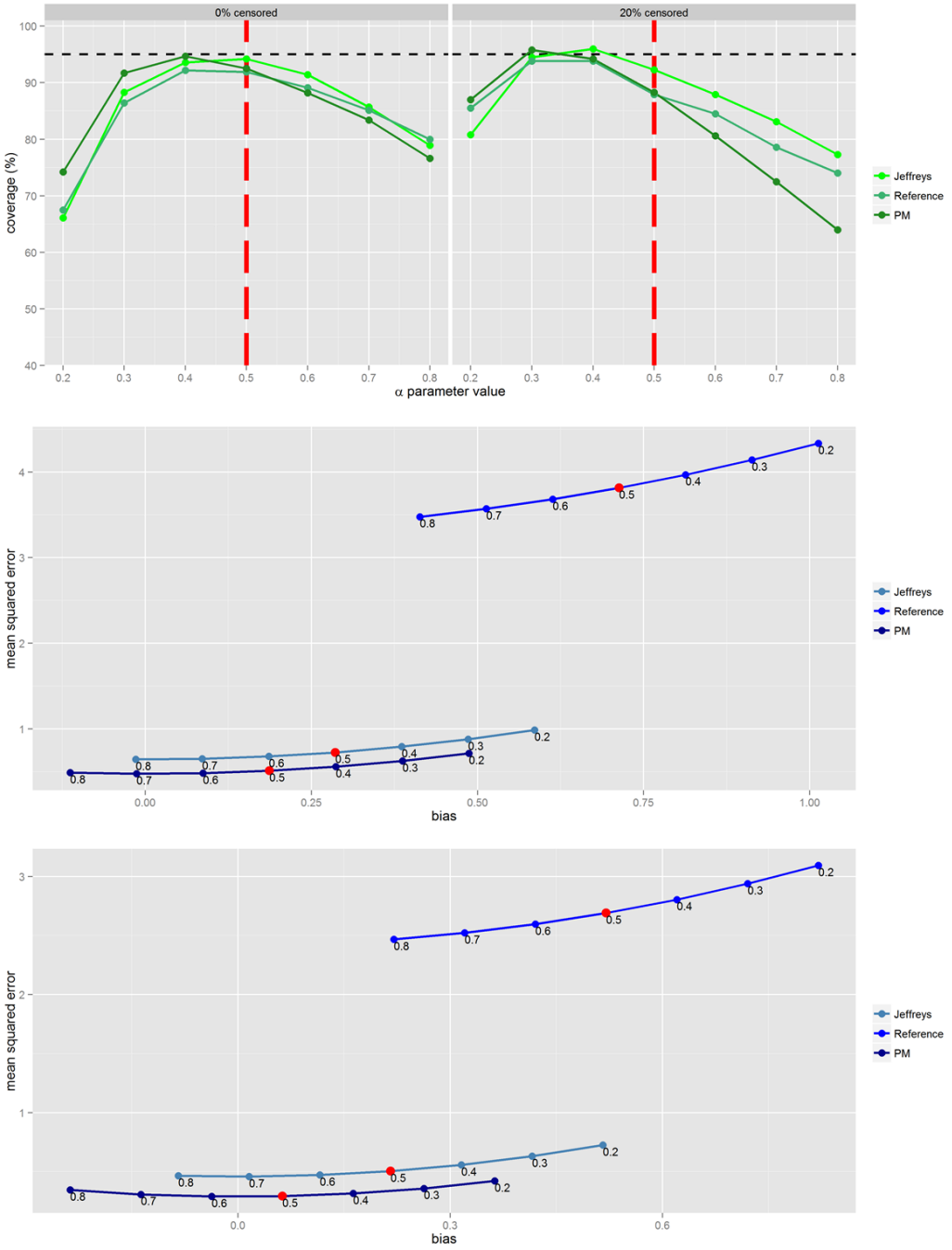


Figure 5.32: *Coverage plots (top) as well as MSE vs bias plots (bottom two) for GCRG model, with* $\alpha = 0.5$ *(*$\beta = 1.5$ *and* $c = 3$*), for all prior distributions and* $\delta = 1$ *(no censoring) and* $\delta = 0.8$ *(20% censored values), and with* $n = 30$*.*

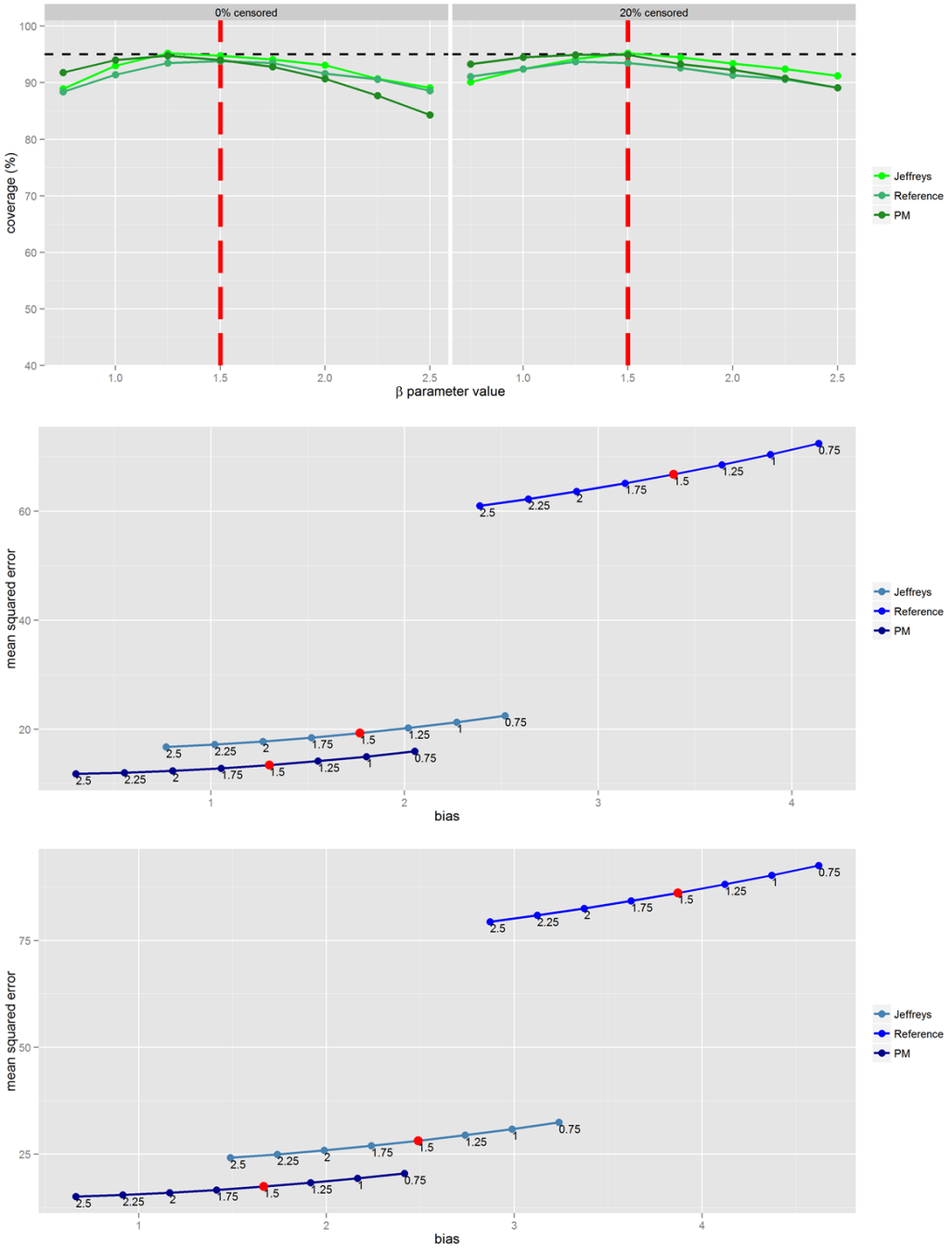


Figure 5.33: *Coverage plots (top) as well as MSE vs bias plots (bottom two) for GCRG model, with* $\beta = 1.5$ *(*$\alpha = 0.5$ *and* $c = 3$*), for all prior distributions and* $\delta = 1$ *(no censoring) and* $\delta = 0.8$ *(20% censored values), and with* $n = 30$*.*

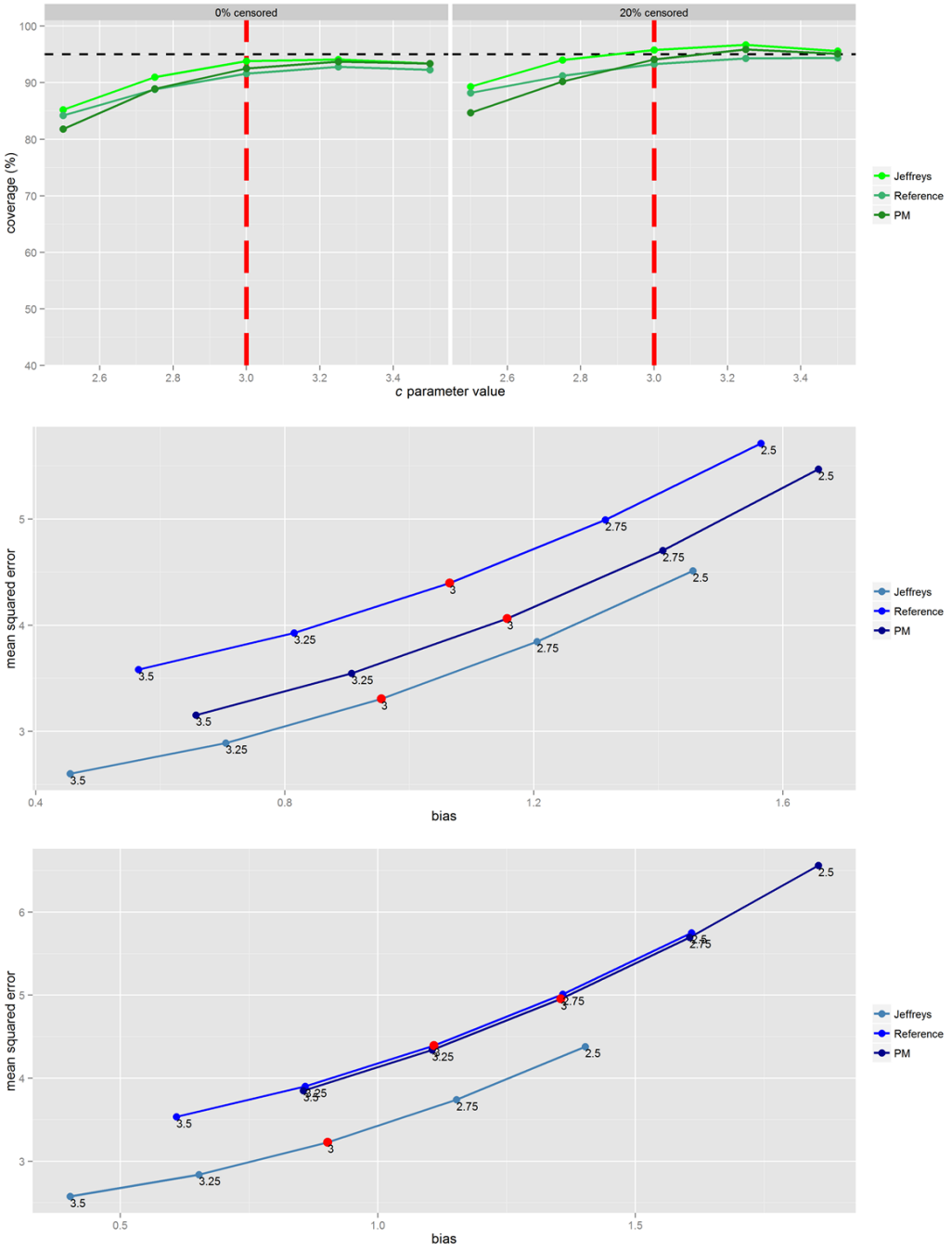


Figure 5.34: *Coverage plots (top) as well as MSE vs bias plots (bottom two) for GCRG model, with* $c = 3$ *(*$\alpha = 0.5$ *and* $\beta = 1.5$*), for all prior distributions and* $\delta = 1$ *(no censoring) and* $\delta = 0.8$ *(20% censored values), and with* $n = 30$*.*

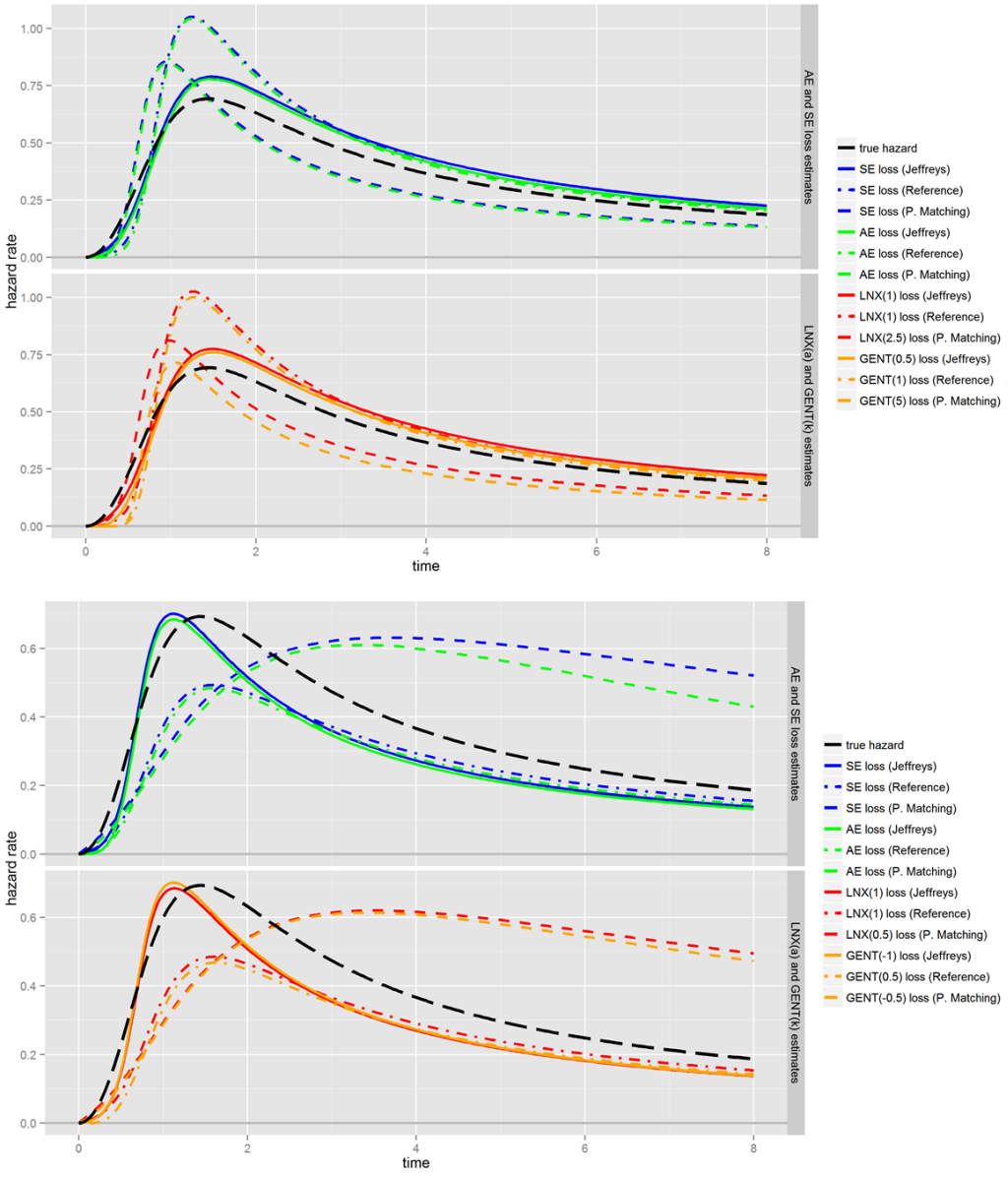


Figure 5.35: *Plots of Bayesian estimates of the hazard function for the GCRG model with* $(\alpha, \beta, c) = (0.5, 1.5, 3)$ *and two levels of censoring (no censoring, top, and 20% censoring, bottom), derived using three different priors, four different loss functions and with sample size* $n = 30$.

Table 5.9: *The MAE, MSE and bias of estimators for the GCRG model, with parameters* $(\alpha, \beta, c) = (0.5, 1.5, 2)$ *and* $n = 30$.

prior	estimator	MAE	MSE	bias	estimator	MAE	MSE	bias	estimator	MAE	MSE	bias
0% censoring												
Jeffreys	$\hat{\alpha}_{\mathrm{AE}}$	0.269	0.192	0.057	$\hat{\beta}_{\mathrm{AE}}$	1.421	6.473	0.666	$\hat{c}_{\mathrm{AE}}$	1.06	2.417	0.695
	$\hat{\alpha}_{\mathrm{SE}}$	0.432	0.726	0.286	$\hat{\beta}_{\mathrm{SE}}$	2.221	19.275	1.769	$\hat{c}_{\mathrm{SE}}$	1.239	3.307	0.955
	$\hat{\alpha}_{\mathrm{LNX}(2)}$	0.205	0.077	0.012	$\hat{\beta}_{\mathrm{LNX}(2)}$	0.676	0.632	-0.423	$\hat{c}_{\mathrm{LNX}(2)}$	0.57	0.484	-0.148
	$\hat{\alpha}_{\mathrm{GE}(2)}$	0.221	0.076	-0.1	$\hat{\beta}_{\mathrm{GE}(2)}$	1.2	2.072	-0.805	$\hat{c}_{\mathrm{GE}(2)}$	0.821	1.325	0.301
Reference	$\hat{\alpha}_{\mathrm{AE}}$	0.397	0.84	0.174	$\hat{\beta}_{\mathrm{AE}}$	1.983	20.346	1.21	$\hat{c}_{\mathrm{AE}}$	1.202	3.378	0.793
	$\hat{\alpha}_{\mathrm{SE}}$	0.867	3.816	0.713	$\hat{\beta}_{\mathrm{SE}}$	3.842	66.787	3.389	$\hat{c}_{\mathrm{SE}}$	1.375	4.397	1.065
	$\hat{\alpha}_{\mathrm{LNX}(2)}$	0.234	0.109	0.032	$\hat{\beta}_{\mathrm{LNX}(2)}$	0.702	0.671	-0.427	$\hat{c}_{\mathrm{LNX}(2)}$	0.631	0.614	-0.161
	$\hat{\alpha}_{\mathrm{GE}(2)}$	0.246	0.112	-0.089	$\hat{\beta}_{\mathrm{GE}(2)}$	1.27	2.512	-0.819	$\hat{c}_{\mathrm{GE}(2)}$	0.933	1.847	0.343
PM	$\hat{\alpha}_{\mathrm{AE}}$	0.235	0.134	-0.008	$\hat{\beta}_{\mathrm{AE}}$	1.257	4.495	0.348	$\hat{c}_{\mathrm{AE}}$	1.166	3.052	0.894
	$\hat{\alpha}_{\mathrm{SE}}$	0.36	0.512	0.187	$\hat{\beta}_{\mathrm{SE}}$	1.885	13.418	1.303	$\hat{c}_{\mathrm{SE}}$	1.366	4.062	1.157
	$\hat{\alpha}_{\mathrm{LNX}(2)}$	0.189	0.061	-0.034	$\hat{\beta}_{\mathrm{LNX}(2)}$	0.718	0.67	-0.541	$\hat{c}_{\mathrm{LNX}(2)}$	0.559	0.5	-0.032
	$\hat{\alpha}_{\mathrm{GE}(2)}$	0.221	0.067	-0.142	$\hat{\beta}_{\mathrm{GE}(2)}$	1.249	1.9	-0.981	$\hat{c}_{\mathrm{GE}(2)}$	0.882	1.668	0.477
20% censoring												
Jeffreys	$\hat{\alpha}_{\mathrm{AE}}$	0.253	0.128	-0.026	$\hat{\beta}_{\mathrm{AE}}$	1.706	9.64	0.987	$\hat{c}_{\mathrm{AE}}$	1.041	2.327	0.62
	$\hat{\alpha}_{\mathrm{SE}}$	0.41	0.505	0.216	$\hat{\beta}_{\mathrm{SE}}$	2.89	28.149	2.488	$\hat{c}_{\mathrm{SE}}$	1.213	3.228	0.902
	$\hat{\alpha}_{\mathrm{LNX}(2)}$	0.2	0.061	-0.052	$\hat{\beta}_{\mathrm{LNX}(2)}$	0.671	0.62	-0.419	$\hat{c}_{\mathrm{LNX}(2)}$	0.58	0.468	-0.277
	$\hat{\alpha}_{\mathrm{GE}(2)}$	0.236	0.072	-0.177	$\hat{\beta}_{\mathrm{GE}(2)}$	1.269	2.155	-0.881	$\hat{c}_{\mathrm{GE}(2)}$	0.798	1.14	0.169
Reference	$\hat{\alpha}_{\mathrm{AE}}$	0.376	0.73	0.057	$\hat{\beta}_{\mathrm{AE}}$	2.288	30.472	1.45	$\hat{c}_{\mathrm{AE}}$	1.218	3.193	0.791
	$\hat{\alpha}_{\mathrm{SE}}$	0.735	2.691	0.521	$\hat{\beta}_{\mathrm{SE}}$	4.334	86.126	3.872	$\hat{c}_{\mathrm{SE}}$	1.425	4.393	1.109
	$\hat{\alpha}_{\mathrm{LNX}(2)}$	0.231	0.096	-0.055	$\hat{\beta}_{\mathrm{LNX}(2)}$	0.729	0.717	-0.489	$\hat{c}_{\mathrm{LNX}(2)}$	0.618	0.568	-0.251
	$\hat{\alpha}_{\mathrm{GE}(2)}$	0.274	0.119	-0.184	$\hat{\beta}_{\mathrm{GE}(2)}$	1.365	3	-0.931	$\hat{c}_{\mathrm{GE}(2)}$	0.897	1.575	0.265
PM	$\hat{\alpha}_{\mathrm{AE}}$	0.254	0.1	-0.112	$\hat{\beta}_{\mathrm{AE}}$	1.427	5.845	0.482	$\hat{c}_{\mathrm{AE}}$	1.298	3.559	1.026
	$\hat{\alpha}_{\mathrm{SE}}$	0.334	0.293	0.063	$\hat{\beta}_{\mathrm{SE}}$	2.211	17.425	1.665	$\hat{c}_{\mathrm{SE}}$	1.551	4.956	1.355
	$\hat{\alpha}_{\mathrm{LNX}(2)}$	0.214	0.062	-0.12	$\hat{\beta}_{\mathrm{LNX}(2)}$	0.734	0.691	-0.571	$\hat{c}_{\mathrm{LNX}(2)}$	0.547	0.47	-0.072
	$\hat{\alpha}_{\mathrm{GE}(2)}$	0.266	0.083	-0.231	$\hat{\beta}_{\mathrm{GE}(2)}$	1.304	1.967	-1.088	$\hat{c}_{\mathrm{GE}(2)}$	0.919	1.693	0.499

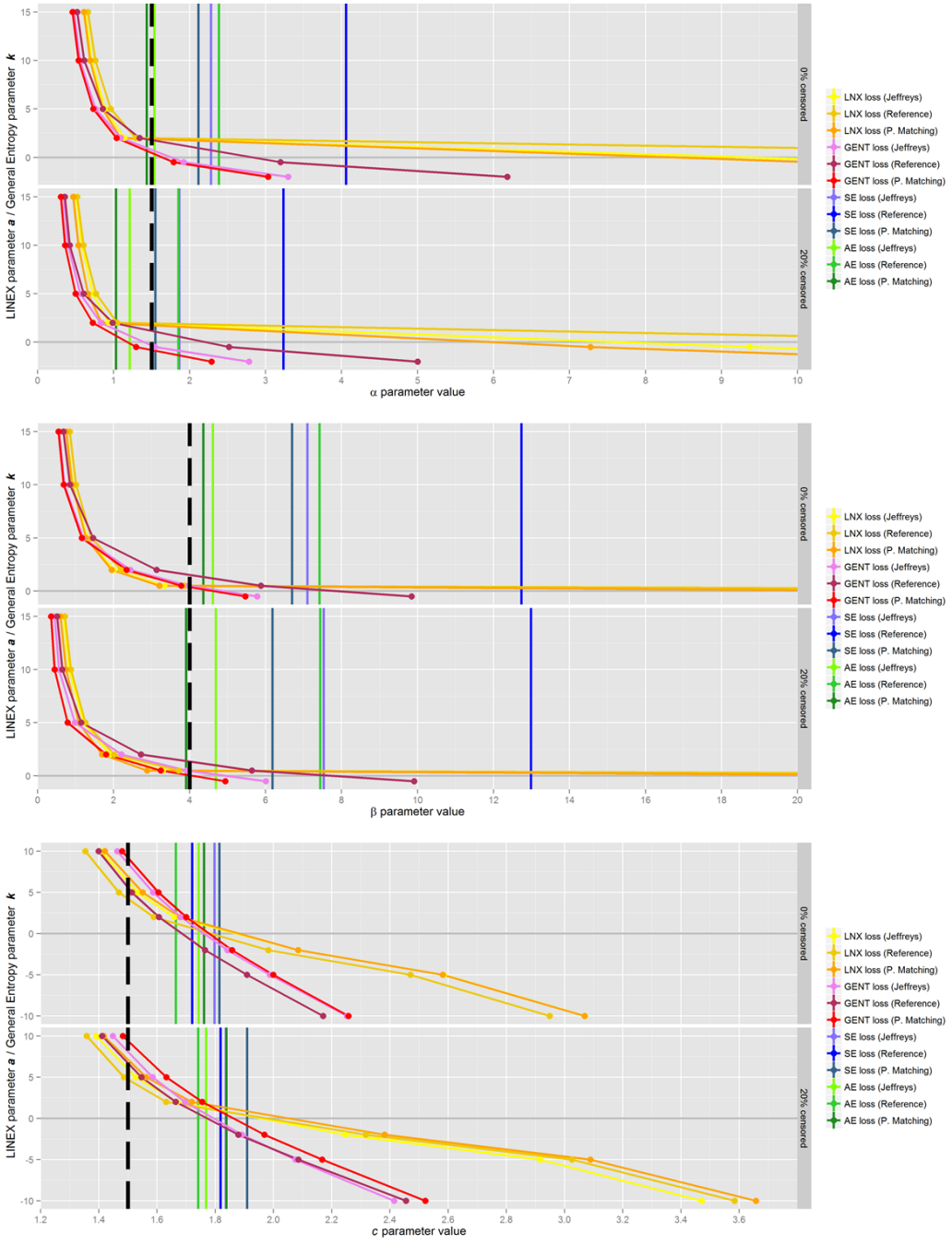


Figure 5.36: *Bayesian estimates plot for GCRG model, with* $\alpha = 1.5$ *(top),* $\beta = 4$ *(middle) and* $c = 1.5$ *(bottom), and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours).*

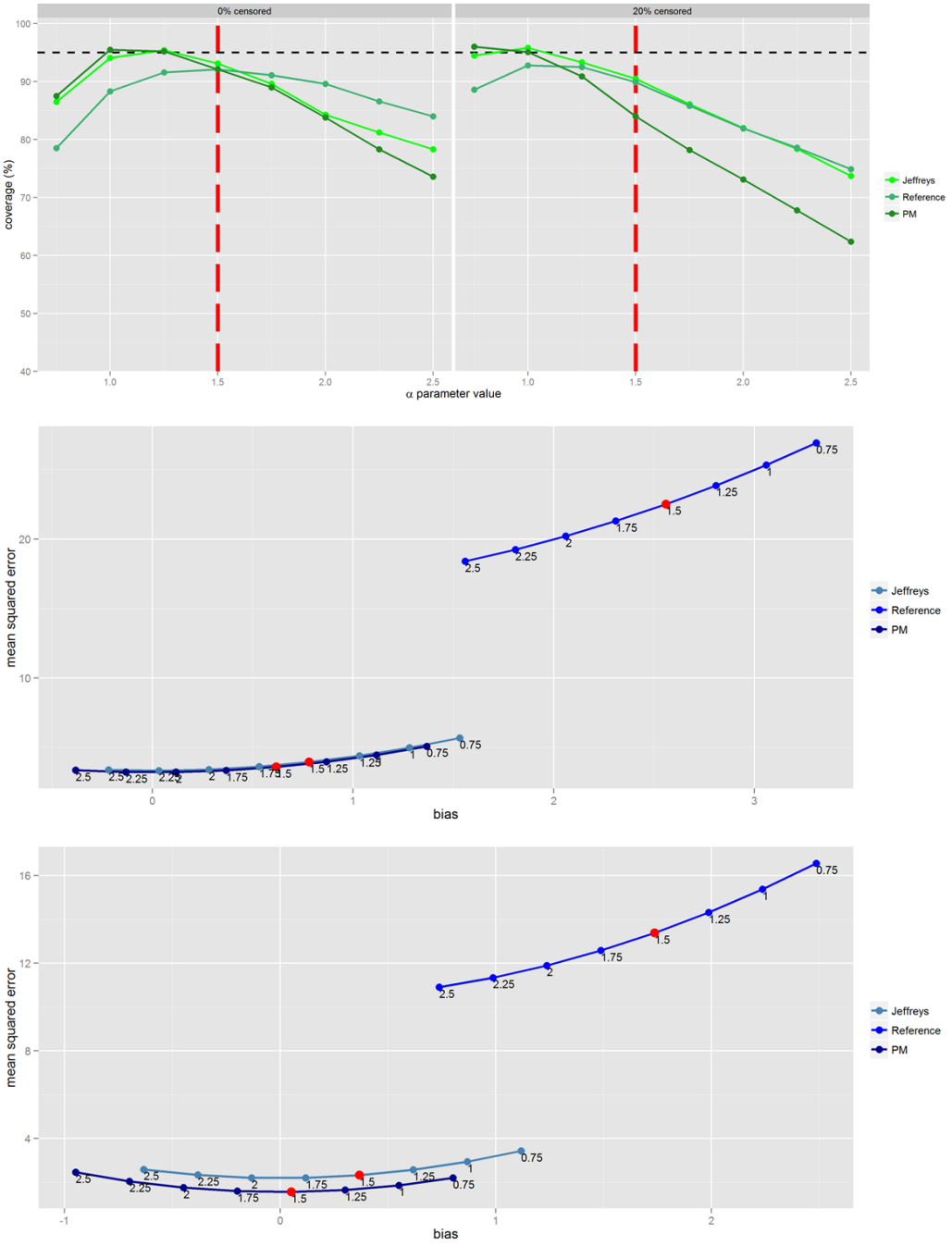


Figure 5.37: *Coverage plots (top) as well as MSE vs bias plots (bottom two) for GCRG model, with* $\alpha = 1.5$ *(*$\beta = 4$ *and* $c = 1.5$*), for all prior distributions and* $\delta = 1$ *(no censoring) and* $\delta = 0.8$ *(20% censored values).*

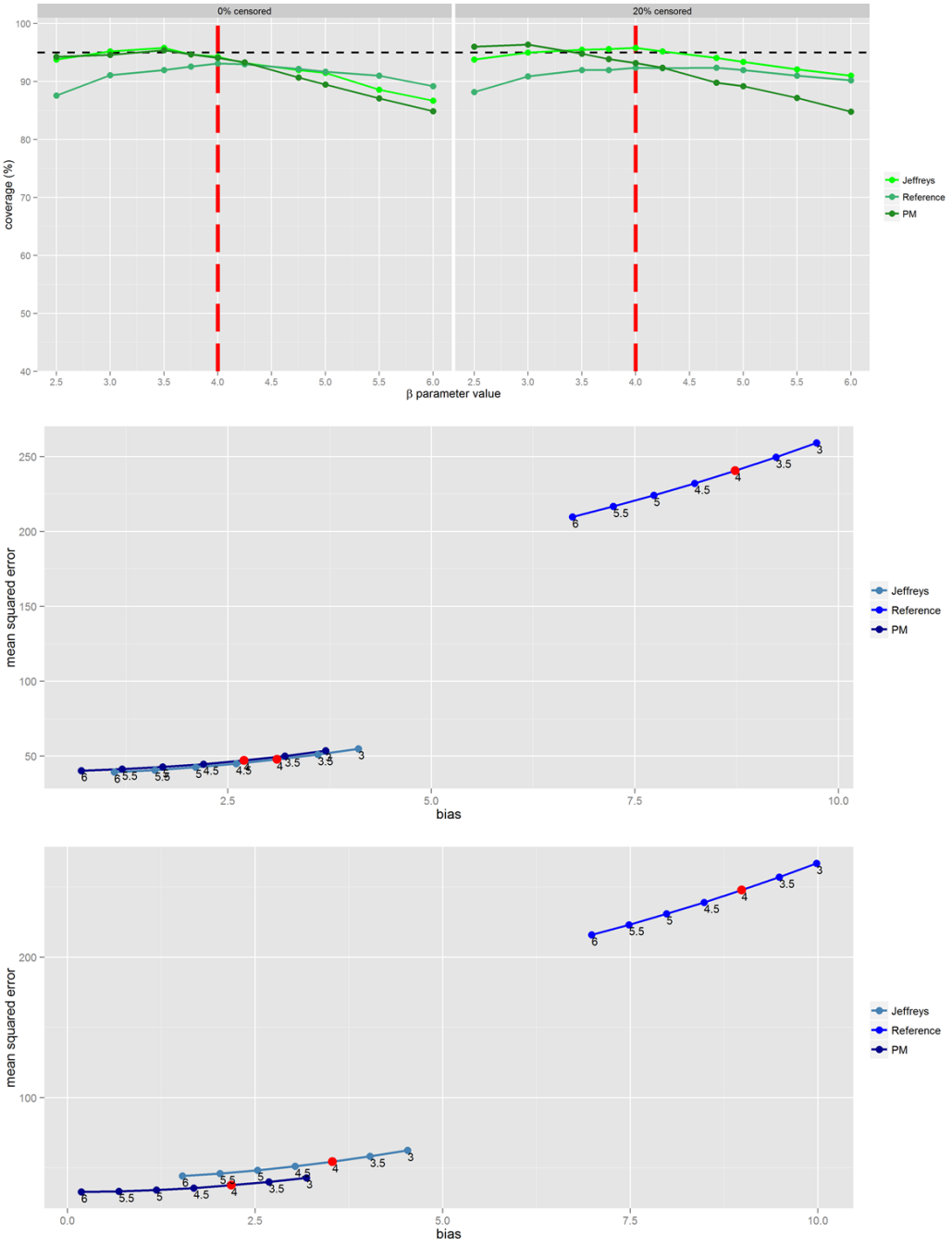


Figure 5.38: *Coverage plots (top) as well as MSE vs bias plots (bottom two) for GCRG model, with $\beta = 4$ ($\alpha = 1.5$ and $c = 1.5$), for all prior distributions and $\delta = 1$ (no censoring) and $\delta = 0.8$ (20% censored values).*

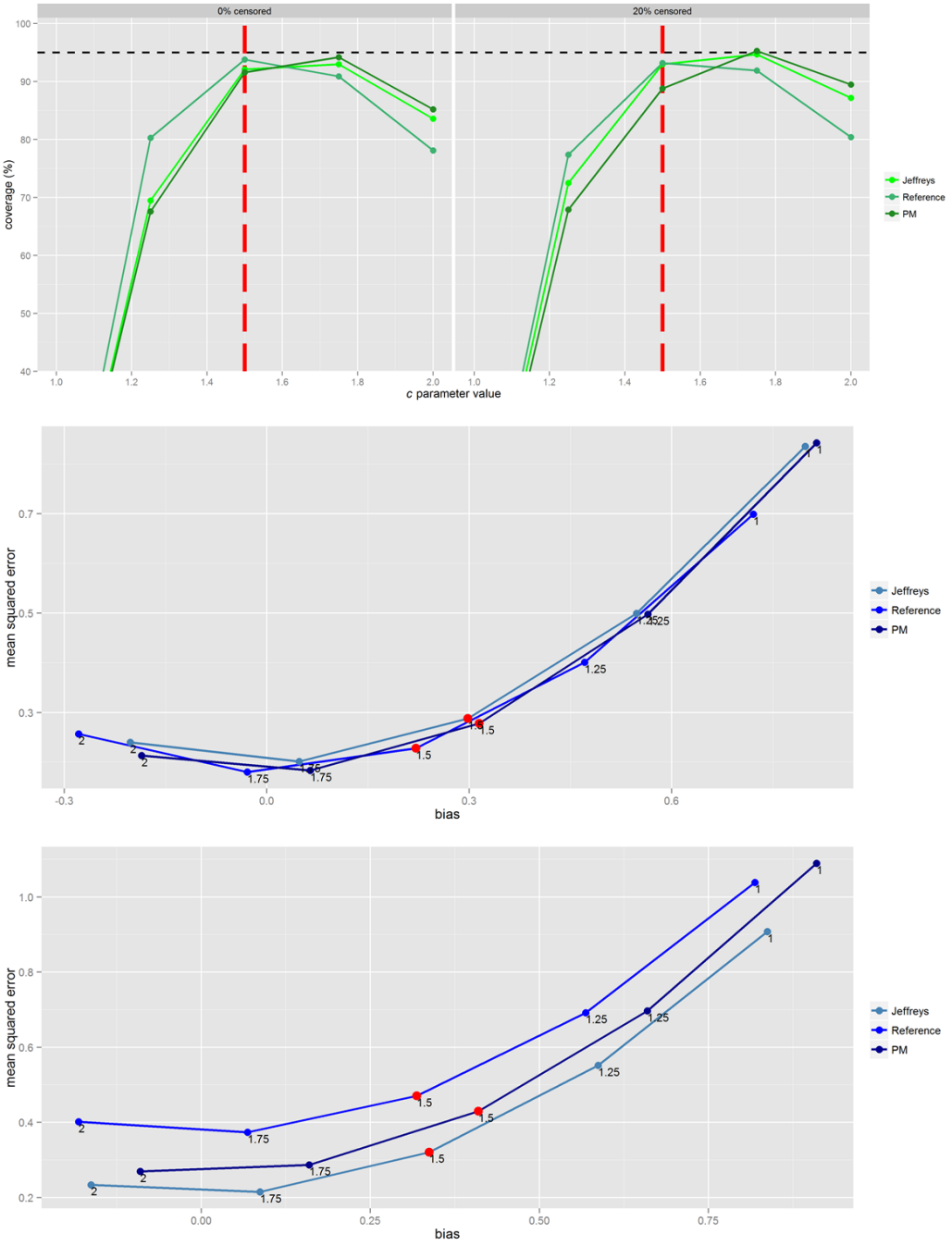


Figure 5.39: *Coverage plots (top) as well as MSE vs bias plots (bottom two) for GCRG model, with $c = 1.5$ ($\alpha = 1.5$ and $\beta = 4$), for all prior distributions and $\delta = 1$ (no censoring) and $\delta = 0.8$ (20% censored values).*

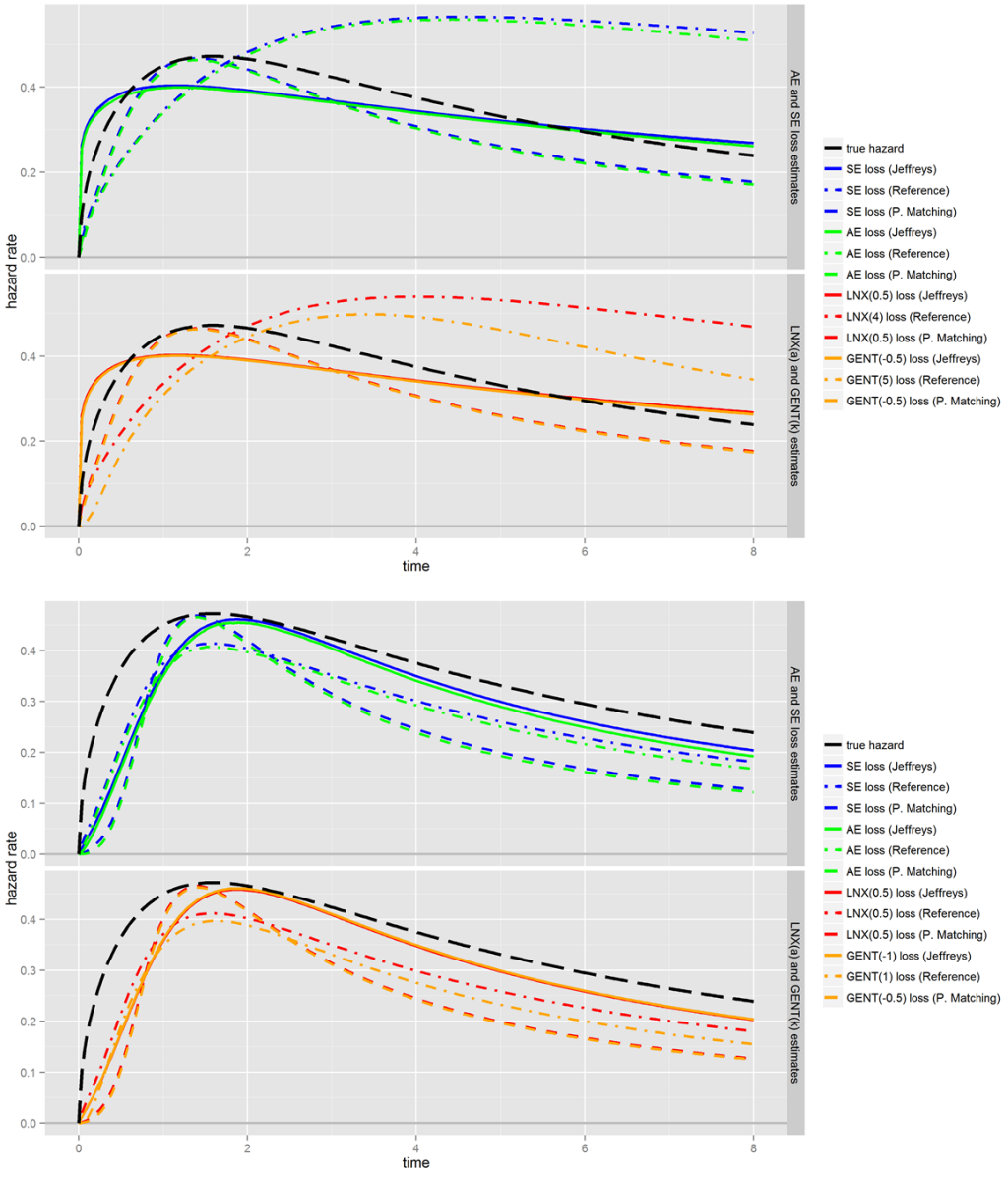


Figure 5.40: *Plots of Bayesian estimates of the hazard function for the GCRG model with* $(\alpha, \beta, c) = (1.5, 4, 1.5)$ *and two levels of censoring (no censoring, top, and 20% censoring, bottom), derived using three different priors and four different loss functions.*

Table 5.10: *The MAE, MSE and bias of estimators for the GCRG model, with parameters* $(\alpha, \beta, c) = (1.5, 4, 1.5)$.

prior	estimator	MAE	MSE	bias	estimator	MAE	MSE	bias	estimator	MAE	MSE	bias
0% censoring												
Jeffreys	$\hat{\alpha}_{\text{AE}}$	0.726	1.079	0.038	$\hat{\beta}_{\text{AE}}$	2.496	14.055	0.614	$\hat{c}_{\text{AE}}$	0.321	0.235	0.243
	$\hat{\alpha}_{\text{SE}}$	1.246	3.948	0.782	$\hat{\beta}_{\text{SE}}$	4.35	47.795	3.104	$\hat{c}_{\text{SE}}$	0.356	0.288	0.298
	$\hat{\alpha}_{\text{LNX(2)}}$	0.497	0.338	-0.36	$\hat{\beta}_{\text{LNX(2)}}$	2.023	4.754	-1.996	$\hat{c}_{\text{LNX(2)}}$	0.25	0.131	0.159
	$\hat{\alpha}_{\text{GE(2)}}$	0.617	0.514	-0.406	$\hat{\beta}_{\text{GE(2)}}$	2.199	6.527	-1.557	$\hat{c}_{\text{GE(2)}}$	0.274	0.17	0.182
Reference	$\hat{\alpha}_{\text{AE}}$	1.46	7.701	0.888	$\hat{\beta}_{\text{AE}}$	5.024	89.13	3.42	$\hat{c}_{\text{AE}}$	0.298	0.19	0.165
	$\hat{\alpha}_{\text{SE}}$	2.92	22.52	2.559	$\hat{\beta}_{\text{SE}}$	9.732	240.635	8.733	$\hat{c}_{\text{SE}}$	0.321	0.228	0.221
	$\hat{\alpha}_{\text{LNX(2)}}$	0.523	0.392	-0.21	$\hat{\beta}_{\text{LNX(2)}}$	1.903	4.366	-1.804	$\hat{c}_{\text{LNX(2)}}$	0.231	0.105	0.088
	$\hat{\alpha}_{\text{GE(2)}}$	0.732	0.93	-0.158	$\hat{\beta}_{\text{GE(2)}}$	2.539	10.581	-0.866	$\hat{c}_{\text{GE(2)}}$	0.252	0.135	0.106
PM	$\hat{\alpha}_{\text{AE}}$	0.713	0.974	-0.064	$\hat{\beta}_{\text{AE}}$	2.454	13.45	0.365	$\hat{c}_{\text{AE}}$	0.332	0.23	0.262
	$\hat{\alpha}_{\text{SE}}$	1.159	3.598	0.617	$\hat{\beta}_{\text{SE}}$	4.113	47.164	2.7	$\hat{c}_{\text{SE}}$	0.366	0.278	0.315
	$\hat{\alpha}_{\text{LNX(2)}}$	0.526	0.364	-0.409	$\hat{\beta}_{\text{LNX(2)}}$	2.08	4.993	-2.052	$\hat{c}_{\text{LNX(2)}}$	0.259	0.133	0.178
	$\hat{\alpha}_{\text{GE(2)}}$	0.644	0.539	-0.462	$\hat{\beta}_{\text{GE(2)}}$	2.304	6.786	-1.662	$\hat{c}_{\text{GE(2)}}$	0.283	0.169	0.2
20% censoring												
Jeffreys	$\hat{\alpha}_{\text{AE}}$	0.711	0.752	-0.287	$\hat{\beta}_{\text{AE}}$	2.557	14.746	0.694	$\hat{c}_{\text{AE}}$	0.35	0.253	0.268
	$\hat{\alpha}_{\text{SE}}$	1.029	2.324	0.367	$\hat{\beta}_{\text{SE}}$	4.651	54.335	3.532	$\hat{c}_{\text{SE}}$	0.395	0.321	0.337
	$\hat{\alpha}_{\text{LNX(2)}}$	0.613	0.474	-0.561	$\hat{\beta}_{\text{LNX(2)}}$	2.112	5.127	-2.093	$\hat{c}_{\text{LNX(2)}}$	0.262	0.134	0.169
	$\hat{\alpha}_{\text{GE(2)}}$	0.743	0.665	-0.669	$\hat{\beta}_{\text{GE(2)}}$	2.395	7.212	-1.791	$\hat{c}_{\text{GE(2)}}$	0.291	0.175	0.196
Reference	$\hat{\alpha}_{\text{AE}}$	1.257	4.266	0.355	$\hat{\beta}_{\text{AE}}$	5.278	88.852	3.439	$\hat{c}_{\text{AE}}$	0.38	0.37	0.242
	$\hat{\alpha}_{\text{SE}}$	2.323	13.383	1.737	$\hat{\beta}_{\text{SE}}$	10.069	247.891	8.984	$\hat{c}_{\text{SE}}$	0.417	0.47	0.319
	$\hat{\alpha}_{\text{LNX(2)}}$	0.617	0.496	-0.453	$\hat{\beta}_{\text{LNX(2)}}$	2.065	5.075	-1.987	$\hat{c}_{\text{LNX(2)}}$	0.277	0.159	0.131
	$\hat{\alpha}_{\text{GE(2)}}$	0.795	0.834	-0.514	$\hat{\beta}_{\text{GE(2)}}$	2.77	11.375	-1.277	$\hat{c}_{\text{GE(2)}}$	0.314	0.245	0.164
PM	$\hat{\alpha}_{\text{AE}}$	0.709	0.696	-0.467	$\hat{\beta}_{\text{AE}}$	2.397	12.119	-0.09	$\hat{c}_{\text{AE}}$	0.404	0.343	0.338
	$\hat{\alpha}_{\text{SE}}$	0.89	1.563	0.051	$\hat{\beta}_{\text{SE}}$	3.855	37.543	2.184	$\hat{c}_{\text{SE}}$	0.456	0.43	0.41
	$\hat{\alpha}_{\text{LNX(2)}}$	0.681	0.564	-0.658	$\hat{\beta}_{\text{LNX(2)}}$	2.317	5.981	-2.304	$\hat{c}_{\text{LNX(2)}}$	0.296	0.172	0.219
	$\hat{\alpha}_{\text{GE(2)}}$	0.806	0.753	-0.774	$\hat{\beta}_{\text{GE(2)}}$	2.536	7.976	-2.197	$\hat{c}_{\text{GE(2)}}$	0.334	0.232	0.255

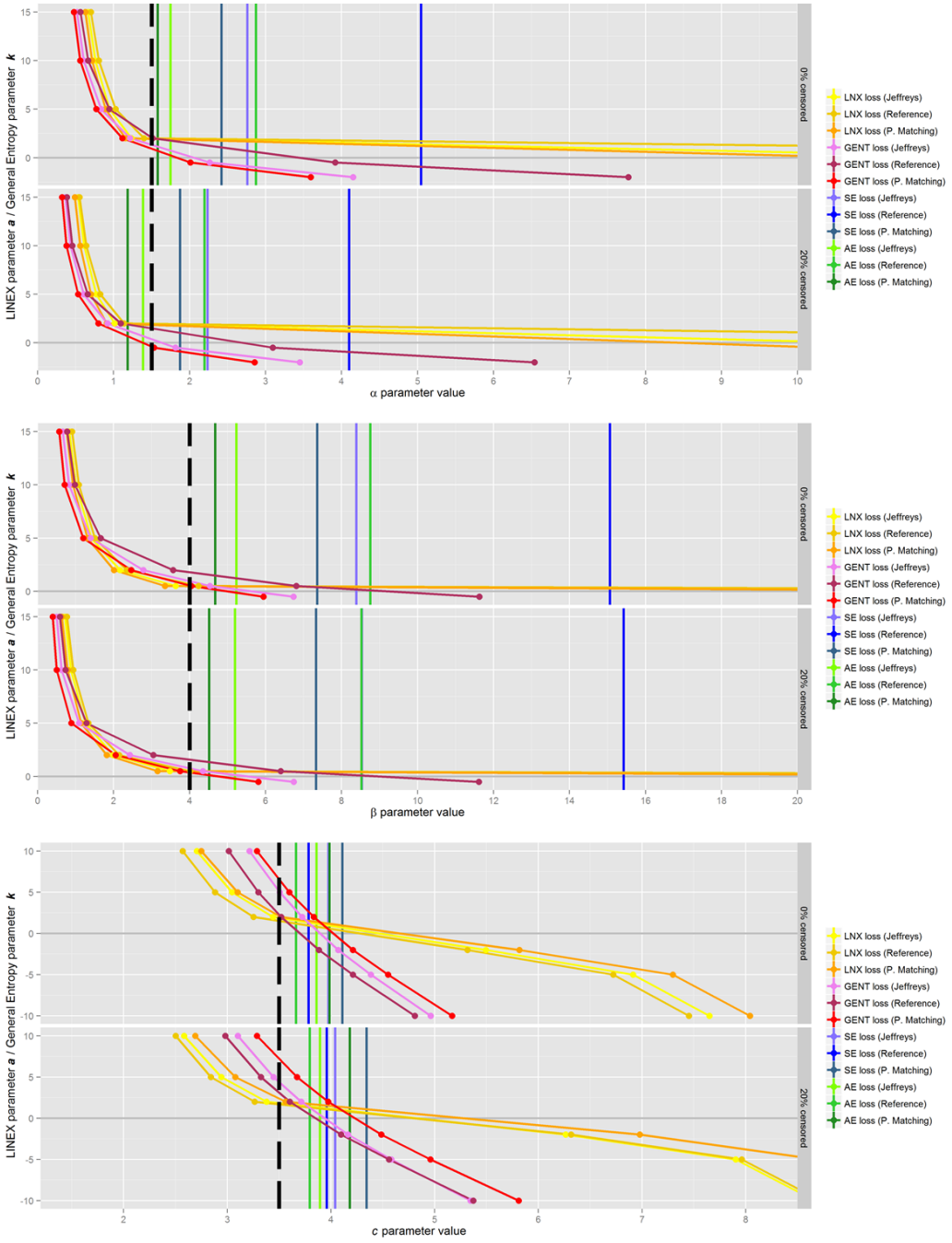


Figure 5.41: *Bayesian estimates plot for GCRG model, with* $\alpha = 1.5$ *(top),* $\beta = 4$ *(middle) and* $c = 3.5$ *(bottom), and* $\delta = 1$ *(no censoring, lighter colours) and* $\delta = 0.8$ *(20% censored values, darker colours).*

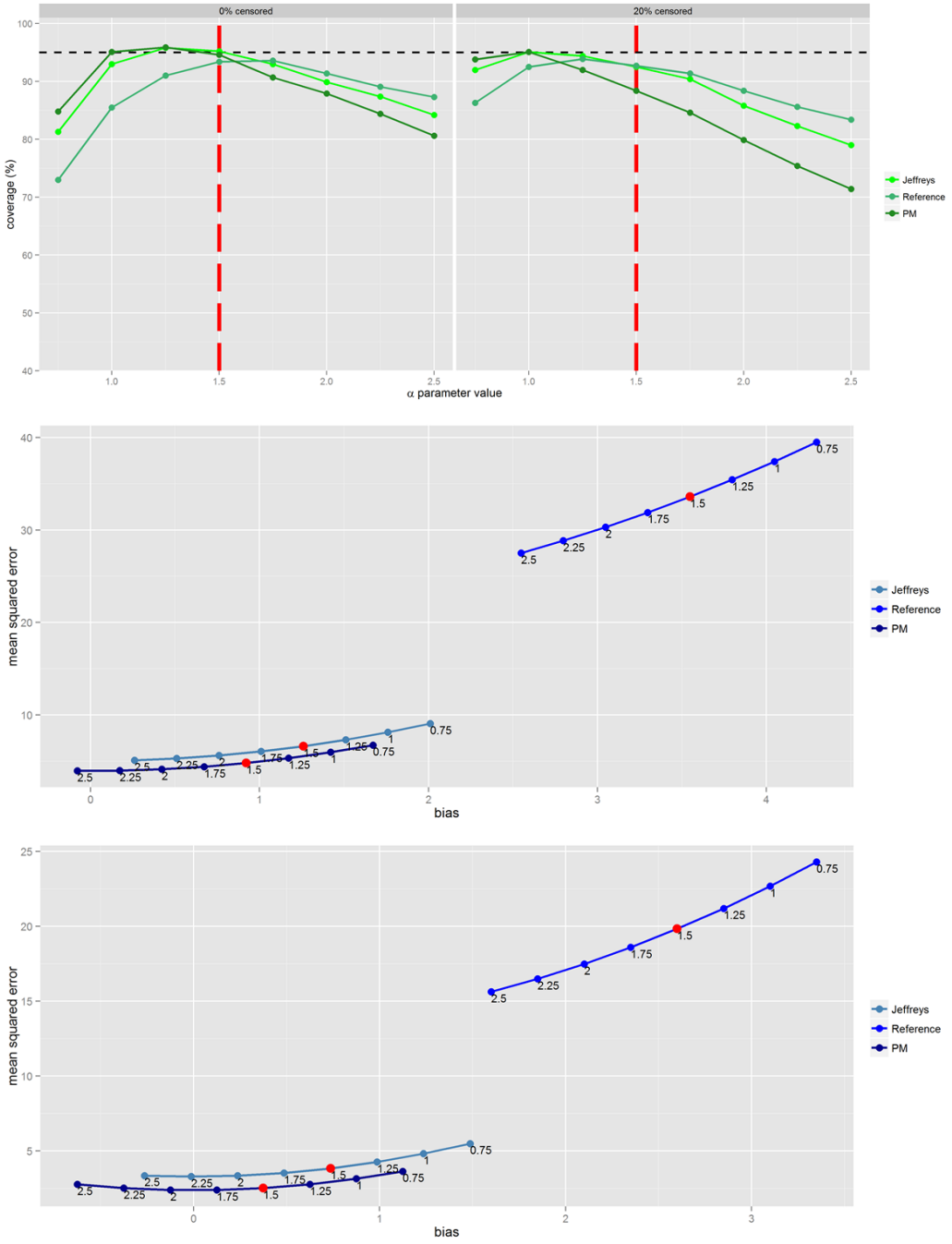


Figure 5.42: *Coverage plots (top) as well as MSE vs bias plots (bottom two) for GCRG model, with* $\alpha = 1.5$ *(*$\beta = 4$ *and* $c = 3.5$*), for all prior distributions and* $\delta = 1$ *(no censoring) and* $\delta = 0.8$ *(20% censored values).*

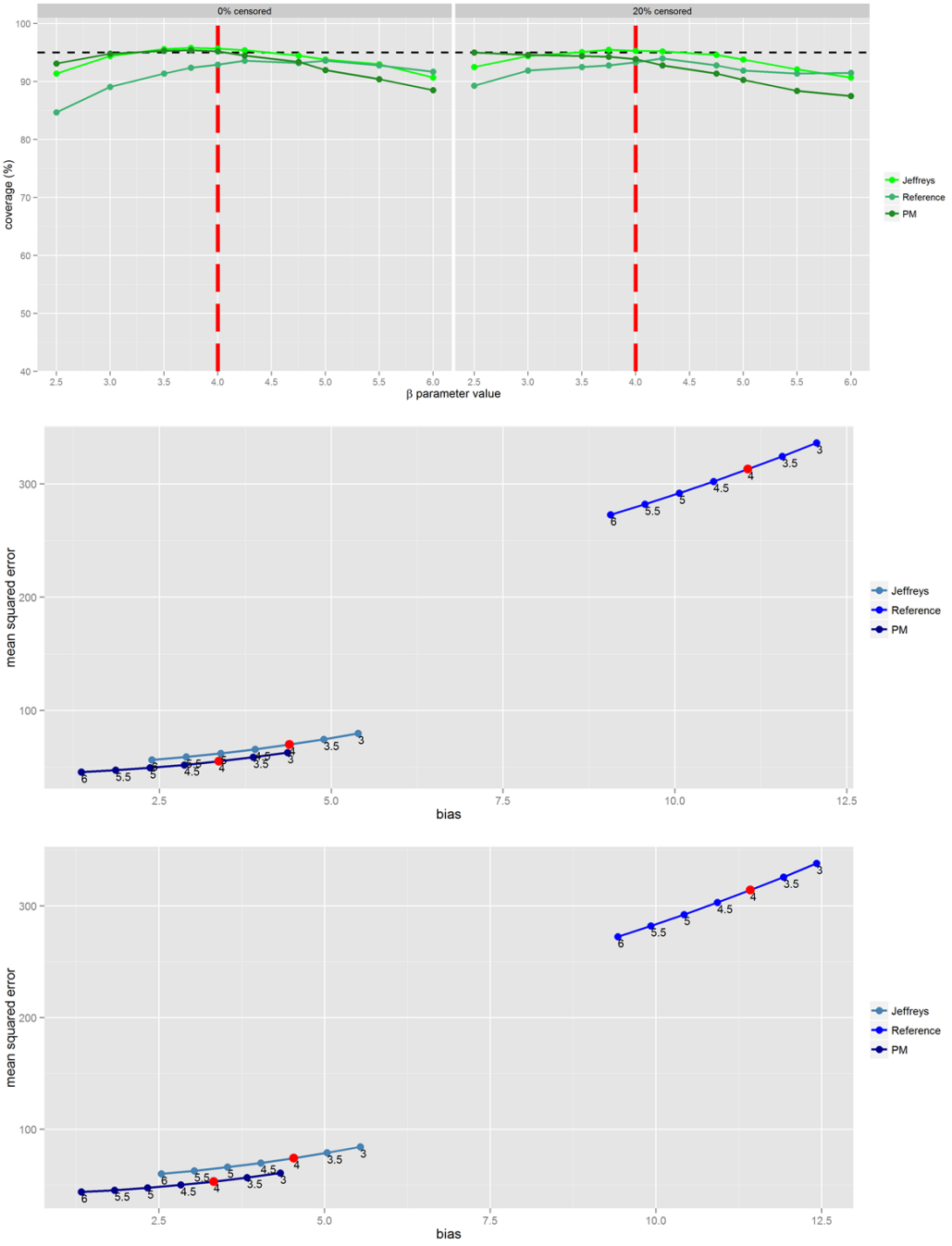


Figure 5.43: *Coverage plots (top) as well as MSE vs bias plots (bottom two) for GCRG model, with $\beta = 4$ ($\alpha = 1.5$ and $c = 3.5$), for all prior distributions and $\delta = 1$ (no censoring) and $\delta = 0.8$ (20% censored values).*

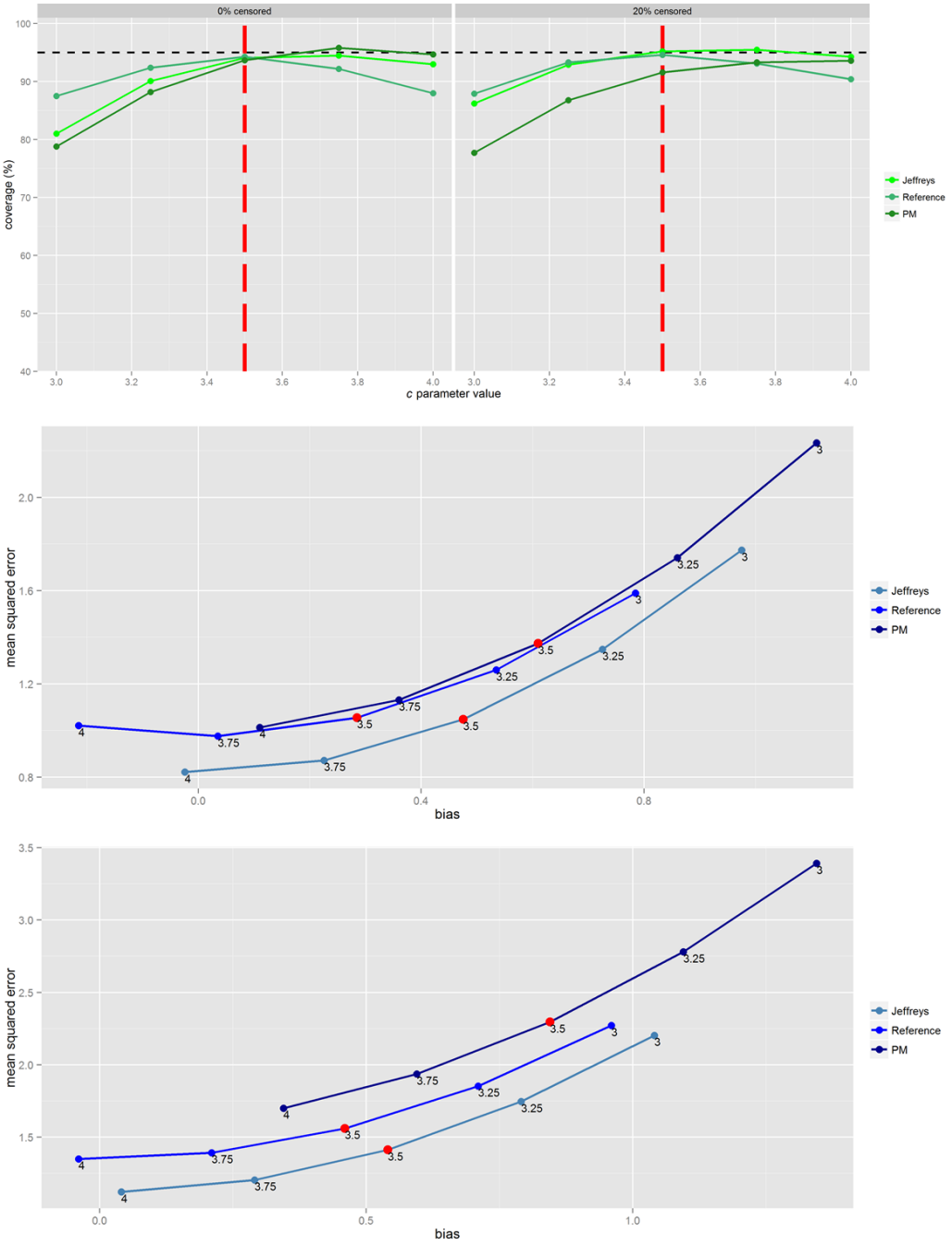


Figure 5.44: *Coverage plots (top) as well as MSE vs bias plots (bottom two) for GCRG model, with* $c = 3.5$ *(*$\alpha = 1.5$ *and* $\beta = 4$*), for all prior distributions and* $\delta = 1$ *(no censoring) and* $\delta = 0.8$ *(20% censored values).*

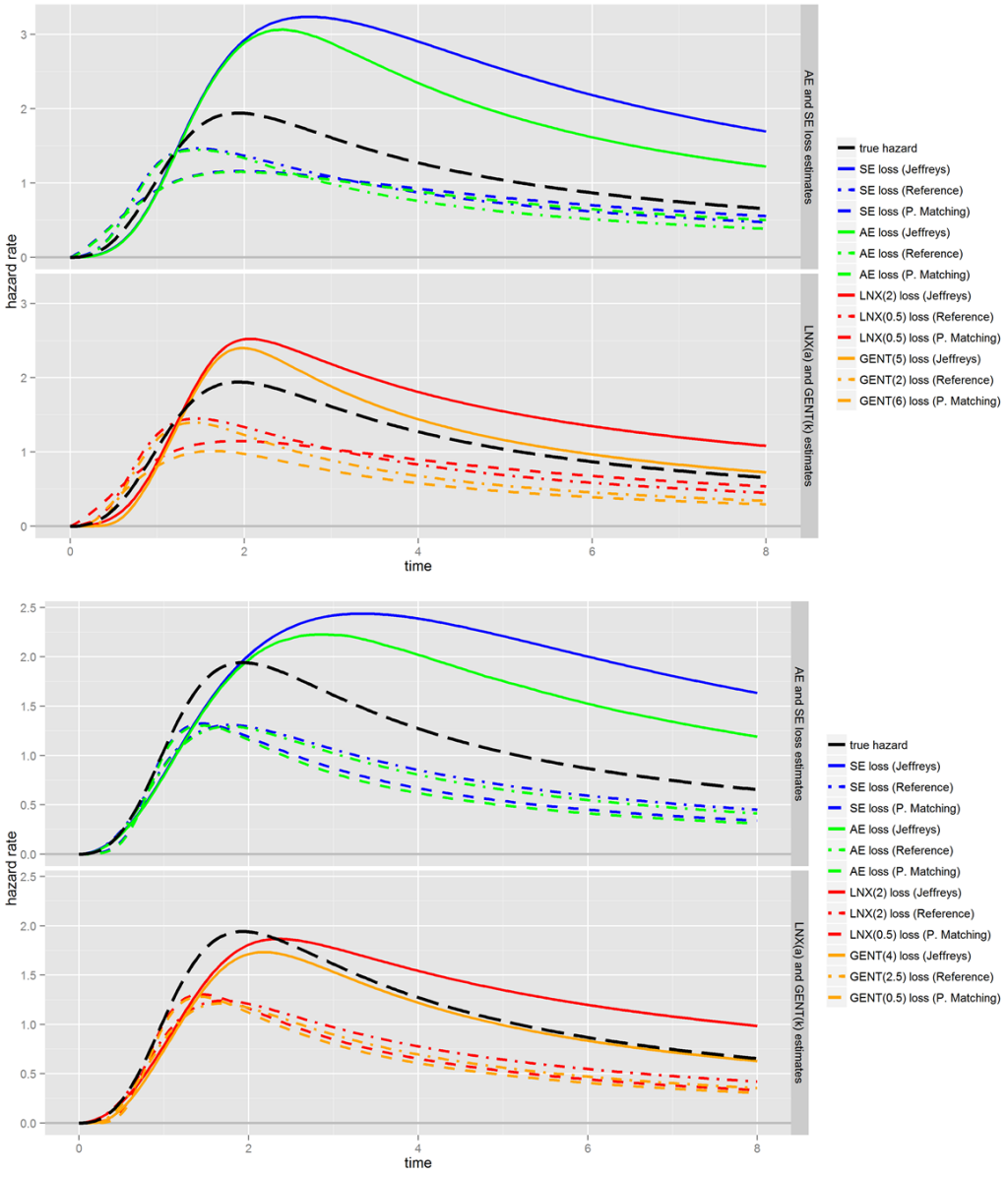


Figure 5.45: *Plots of Bayesian estimates of the hazard function for the GCRG model with* $(\alpha, \beta, c) = (1.5, 4, 3.5)$ *and two levels of censoring (no censoring, top, and 20% censoring, bottom), derived using three different priors and four different loss functions.*

Table 5.11: *The MAE, MSE and bias of estimators for the GCRG model, with parameters* $(\alpha, \beta, c) = (1.5, 4, 3.5)$.

prior	estimator	MAE	MSE	bias	estimator	MAE	MSE	bias	estimator	MAE	MSE	bias
					0% censoring							
Jeffreys	$\hat{\alpha}_{\mathrm{AE}}$	0.809	1.503	0.25	$\hat{\beta}_{\mathrm{AE}}$	2.747	18.56	1.232	$\hat{c}_{\mathrm{AE}}$	0.641	0.86	0.361
	$\hat{\alpha}_{\mathrm{SE}}$	1.61	6.627	1.26	$\hat{\beta}_{\mathrm{SE}}$	5.306	69.925	4.391	$\hat{c}_{\mathrm{SE}}$	0.701	1.048	0.476
	$\hat{\alpha}_{\mathrm{LNX}(2)}$	0.461	0.298	-0.266	$\hat{\beta}_{\mathrm{LNX}(2)}$	1.874	4.172	-1.833	$\hat{c}_{\mathrm{LNX}(2)}$	0.448	0.327	-0.058
	$\hat{\alpha}_{\mathrm{GE}(2)}$	0.594	0.512	-0.285	$\hat{\beta}_{\mathrm{GE}(2)}$	2.131	6.277	-1.22	$\hat{c}_{\mathrm{GE}(2)}$	0.569	0.648	0.223
Reference	$\hat{\alpha}_{\mathrm{AE}}$	1.845	10.803	1.375	$\hat{\beta}_{\mathrm{AE}}$	6.093	114.295	4.757	$\hat{c}_{\mathrm{AE}}$	0.645	0.886	0.164
	$\hat{\alpha}_{\mathrm{SE}}$	3.827	33.607	3.548	$\hat{\beta}_{\mathrm{SE}}$	11.819	313.054	11.063	$\hat{c}_{\mathrm{SE}}$	0.682	1.055	0.285
	$\hat{\alpha}_{\mathrm{LNX}(2)}$	0.524	0.411	-0.099	$\hat{\beta}_{\mathrm{LNX}(2)}$	1.784	3.971	-1.651	$\hat{c}_{\mathrm{LNX}(2)}$	0.519	0.394	-0.247
	$\hat{\alpha}_{\mathrm{GE}(2)}$	0.782	1.171	0.015	$\hat{\beta}_{\mathrm{GE}(2)}$	2.629	12.578	-0.433	$\hat{c}_{\mathrm{GE}(2)}$	0.587	0.665	0.021
PM	$\hat{\alpha}_{\mathrm{AE}}$	0.761	1.217	0.08	$\hat{\beta}_{\mathrm{AE}}$	2.555	15.583	0.677	$\hat{c}_{\mathrm{AE}}$	0.724	1.117	0.488
	$\hat{\alpha}_{\mathrm{SE}}$	1.37	4.807	0.921	$\hat{\beta}_{\mathrm{SE}}$	4.535	55.131	3.363	$\hat{c}_{\mathrm{SE}}$	0.802	1.374	0.61
	$\hat{\alpha}_{\mathrm{LNX}(2)}$	0.498	0.335	-0.346	$\hat{\beta}_{\mathrm{LNX}(2)}$	2.02	4.716	-1.989	$\hat{c}_{\mathrm{LNX}(2)}$	0.45	0.339	0.018
	$\hat{\alpha}_{\mathrm{GE}(2)}$	0.622	0.527	-0.386	$\hat{\beta}_{\mathrm{GE}(2)}$	2.233	6.753	-1.535	$\hat{c}_{\mathrm{GE}(2)}$	0.628	0.803	0.333
					20% censoring							
Jeffreys	$\hat{\alpha}_{\mathrm{AE}}$	0.751	1.092	-0.111	$\hat{\beta}_{\mathrm{AE}}$	2.964	21.13	1.196	$\hat{c}_{\mathrm{AE}}$	0.714	1.121	0.394
	$\hat{\alpha}_{\mathrm{SE}}$	1.28	3.826	0.737	$\hat{\beta}_{\mathrm{SE}}$	5.568	74.226	4.535	$\hat{c}_{\mathrm{SE}}$	0.796	1.413	0.541
	$\hat{\alpha}_{\mathrm{LNX}(2)}$	0.575	0.428	-0.487	$\hat{\beta}_{\mathrm{LNX}(2)}$	2.043	4.889	-2.012	$\hat{c}_{\mathrm{LNX}(2)}$	0.468	0.346	-0.122
	$\hat{\alpha}_{\mathrm{GE}(2)}$	0.719	0.643	-0.581	$\hat{\beta}_{\mathrm{GE}(2)}$	2.405	7.642	-1.571	$\hat{c}_{\mathrm{GE}(2)}$	0.618	0.768	0.214
Reference	$\hat{\alpha}_{\mathrm{AE}}$	1.437	5.507	0.698	$\hat{\beta}_{\mathrm{AE}}$	6.029	105.49	4.527	$\hat{c}_{\mathrm{AE}}$	0.768	1.255	0.297
	$\hat{\alpha}_{\mathrm{SE}}$	3.032	19.829	2.6	$\hat{\beta}_{\mathrm{SE}}$	12.282	314.166	11.423	$\hat{c}_{\mathrm{SE}}$	0.837	1.561	0.46
	$\hat{\alpha}_{\mathrm{LNX}(2)}$	0.569	0.438	-0.363	$\hat{\beta}_{\mathrm{LNX}(2)}$	1.954	4.623	-1.858	$\hat{c}_{\mathrm{LNX}(2)}$	0.543	0.436	-0.237
	$\hat{\alpha}_{\mathrm{GE}(2)}$	0.763	0.804	-0.406	$\hat{\beta}_{\mathrm{GE}(2)}$	2.682	12.068	-0.954	$\hat{c}_{\mathrm{GE}(2)}$	0.669	0.858	0.103
PM	$\hat{\alpha}_{\mathrm{AE}}$	0.738	0.795	-0.315	$\hat{\beta}_{\mathrm{AE}}$	2.64	15.14	0.519	$\hat{c}_{\mathrm{AE}}$	0.9	1.841	0.683
	$\hat{\alpha}_{\mathrm{SE}}$	1.093	2.514	0.375	$\hat{\beta}_{\mathrm{SE}}$	4.677	53.274	3.328	$\hat{c}_{\mathrm{SE}}$	1.016	2.296	0.845
	$\hat{\alpha}_{\mathrm{LNX}(2)}$	0.636	0.51	-0.589	$\hat{\beta}_{\mathrm{LNX}(2)}$	2.2	5.48	-2.179	$\hat{c}_{\mathrm{LNX}(2)}$	0.5	0.438	0.062
	$\hat{\alpha}_{\mathrm{GE}(2)}$	0.769	0.705	-0.701	$\hat{\beta}_{\mathrm{GE}(2)}$	2.457	7.685	-1.946	$\hat{c}_{\mathrm{GE}(2)}$	0.753	1.249	0.475

5.2.5 Discussion

The discussion of the GCRG model's simulation results will start by examining the effect of censoring. Evidently, the transition from using non-censored data to 20% censoring does not seem to deter the results much overall. It does have a slight effect on the accuracy and nature of estimation, most noticeably on the α parameter, where censoring causes some underestimation. However, there is little effect on the point estimations for β and c, apart from a minor inflation in their values. Furthermore, an increase in MAE, MSE and bias is observed for all parameters with censoring, as expected. A slight increase in performance is also seen with in increase in sample size in the one case where it was investigated.

The performance of the estimators derived with different loss functions and priors for α and β are largely similar to what was seen for the CRG model in Section 4.2.5. This makes sense, seeing as the priors derived in this case were conditional upon the generalisation model parameter c. For all three model parameters, the estimators derived using the AE loss function were closer to reality than those derived using the SE loss function. The MAE, MSE and bias were also lower for AE-based estimators across all results.

Considering the estimators of the asymmetric loss functions, a few interesting things can be observed. While the shape of the GE related estimates across a range of k values seems hyperbolic, it seems that estimates related to LINEX loss are almost cubic in nature. This is seen especially for $\hat{\alpha}_{\mathrm{LNX(a)}}$ and $\hat{c}_{\mathrm{LNX(a)}}$, although, the hyperbolic shape only changes for relatively small values of a. Furthermore, for both α and β, the values of a and k that yields the true parameters values stay in the same vicinity for different levels of censoring and different parameter configurations. This is not true for c, though, since the values of a and k that correspond to true values increases as the value of c increases. However, there is very little difference with censoring.

Finally, the performance of estimators derived using different prior distributions are compared. For α and β, it is immediately clear that reference prior suffered from extremely poor performance. Although decent coverage levels were attained, in almost all cases these Bayesian point estimates were the least accurate and the MSE and bias very large. The Jeffreys and PM priors both fared much better and have similar performance. In

general, the Jeffreys prior lead to better coverage, especially with 20% censoring, but the PM prior produces estimates with lower MAE, MSE and bias scores than the Jeffreys.

For the generalisation parameter c, the case was different. The three priors showed relatively similar performance across all measures, but it seems that the Jeffreys prior was superior, with most accurate Bayesian estimates and lowest estimator measures of accuracy and precision.

5.3 Summary

This chapter explores the GCRE and GCRG models and their respective simulation studies. The characteristics of these models derived here are summarised in this section.

The likelihood functions that follow are defined for a given sample of n survival times $\mathbf{t} = (t_1, t_2, \ldots, t_n)$ ordered such that the first d are non-censored and the remaining $(n-d)$ right censored.

	GCRE model
survival function	$S(t,\gamma,c) = \left(1+\frac{t^c}{\gamma}\right)^{-1}$
hazard rate	$h(t,\gamma,c) = \frac{ct^{c-1}}{t^c+\gamma}$
Fisher information	$\begin{bmatrix} \frac{1}{3\gamma^2} & -2A_1(\gamma,c) \\ -2A_1(\gamma,c) & \frac{1}{c^2}+2A_2(\gamma,c) \end{bmatrix}$
likelihood	$\mathcal{L}(\gamma,c\|\mathbf{t}) \propto c^d\gamma^n e^{W_1(\gamma,c)-W_2(\gamma,c)}$ where $W_1(\gamma,c) = \sum_{i=1}^{d}\left[(c-1)\ln t_i - \ln(t_i^c+\gamma)\right]$ and $W_2(\gamma,c) = \sum_{i=1}^{n}\ln\left(t_i^c+\gamma\right)$

In the Fisher information matrix model, A_1 and A_2 denotes two insoluble integrals, which will be approximated with an adaptive quadrature routine. They are defined as

$$A_1(\gamma,c) = \mathrm{E}\left[\frac{T^c\ln T}{(T^c+\gamma)^2}\right] = \int_0^\infty c\gamma\frac{t^{2c-1}\ln t}{(t^c+\gamma)^4}dt$$
$$A_2(\gamma,c) = \mathrm{E}\left[\frac{T^c(\ln T)^2}{(T^c+\gamma)^2}\right] = \int_0^\infty c\gamma\frac{t^{2c-1}(\ln t)^2}{(t^c+\gamma)^4}dt.$$

Only the Jeffreys prior was considered for the GCRE model, such that

$$\pi_{\text{jeff}}(\gamma,c) \propto \sqrt{\frac{1}{3\gamma^2}\left(\frac{1}{c^2}+2A_2(\gamma,c)\right) - 4\gamma c\left(A_1(\gamma,c)\right)^2}.$$

	GCRG model
survival function	$S(t,\alpha,\beta,c) = \left(1+\frac{t^c}{\beta}\right)^{-\alpha}$
hazard rate	$h(t,\alpha,\beta,c) = \frac{\alpha c t^{c-1}}{t^c+\beta}$
Fisher information	$\begin{bmatrix} \frac{1}{\alpha^2} & \frac{-1}{\beta(\alpha+1)} & A_1(\alpha,\beta,c) \\ \frac{-1}{\beta(\alpha+1)} & \frac{\alpha}{\beta^2(\alpha+2)} & -(\alpha+1)A_2(\alpha,\beta,c) \\ A_1(\alpha,\beta,c) & -(\alpha+1)A_2(\alpha,\beta,c) & \frac{1}{c^2}+(\alpha+1)\beta A_3(\alpha,\beta,c) \end{bmatrix}$
likelihood	$\mathcal{L}(\alpha,\beta,c\|\mathbf{t}) \propto (c\alpha)^d e^{W_1(\beta,c)-\alpha W_2(\beta,c)}$ where $W_1(\beta,c) = \sum_{i=1}^{d}\left[(c-1)\ln t_i - \ln(t_i^c+\beta)\right]$ and $W_2(\beta,c) = \sum_{i=1}^{n}\ln\left(1+\frac{t_i^c}{\beta}\right)$

In the Fisher information matrix of the GCRE model, A_1, A_2 and A_3 denotes three insoluble integrals, defined as

$$A_1(\alpha,\beta,c) = \mathrm{E}\left[\frac{T^c \ln T}{T^c+\beta}\right] = \int_0^\infty \alpha c\beta^\alpha \frac{t^{2c-1}\ln t}{(t^c+\beta)^{\alpha+2}}dt$$
$$A_2(\alpha,\beta,c) = \mathrm{E}\left[\frac{T^c \ln T}{(T^c+\beta)^2}\right] = \int_0^\infty \alpha c\beta^\alpha \frac{t^{2c-1}\ln t}{(t^c+\beta)^{\alpha+3}}dt$$
$$A_3(\alpha,\beta,c) = \mathrm{E}\left[\frac{T^c (\ln T)^2}{(T^c+\beta)^2}\right] = \int_0^\infty \alpha c\beta^\alpha \frac{t^{2c-1}(\ln t)^2}{(t^c+\beta)^{\alpha+3}}dt.$$

The non-informative priors for the GCRG model were derived by making use of those found for the CRG model. This formulation is advantageous in terms of computational burden, since this means only one unknown integral, A_3, needs to be numerically approximated.

Jeffreys prior	$\pi_{\text{jeff}}(\alpha,\beta,c)$ $= \pi_{\text{jeff}}(\alpha,\beta)\pi_{\text{jeff}}(c\|\alpha,\beta)$ $\propto \frac{1}{\beta(\alpha+1)\sqrt{\alpha(\alpha+2)}} \cdot \sqrt{\frac{1}{c^2}+(\alpha+1)c\alpha\beta^{\alpha+1}A_3}$
reference prior	$\pi_{\text{ref}}(\alpha,\beta,c)$ $= \pi_{\text{ref}}(\alpha,\beta)\pi_{\text{ref}}(c\|\alpha,\beta)$ $\propto \frac{1}{\alpha\beta} \cdot \sqrt{\frac{1}{c^2}+(\alpha+1)c\alpha\beta^{\alpha+1}A_3}$
PM prior	$\pi_{\text{PM}}(\alpha,\beta,c)$ $= \pi_{\text{PM}}(\alpha,\beta)\pi_{\text{PM}}(c\|\alpha,\beta)$ $\propto \frac{1}{\alpha\beta(\alpha+1)} \cdot \sqrt{\frac{1}{c^2}+(\alpha+1)c\alpha\beta^{\alpha+1}A_3}$

The exposition of results for the simulation studies for these models and corresponding discussions can be found in Sections 5.1.4 and 5.1.5 for the GCRE model, and Sections 5.2.4 and 5.2.5 for the GCRG model.

Chapter 6

Illustration and application of the results

6.1 Motivation and methods

In the previous chapters, a comprehensive simulation study was performed to investigate the properties and behaviour of the four compound Rayleigh models, specifically regarding the effects of prior distributions, censoring and various Bayesian estimators. Even though simulation of the survival times is an indispensable tool for this task, it creates a somewhat sterile environment, thus it is a worthwhile endeavour to consider the models and estimators of interest when applied in practice. In this chapter, two data sets are analysed, one regarding wind speeds and the other regarding cancer patients that received chemotherapy treatment.

In the simulation study, true values of model parameters were known, allowing for clear comparisons between estimators derived with different loss functions and different prior distributions, using the MAE and MSE and investigating frequentist properties with the coverage. Here, however, the use of real world data renders these measures of accuracy infeasible and the process of comparing different estimators and models becomes much less straightforward. It should be noted that the comparison of different models is not the primary focus, but rather done for the sake of interest. In the next section, model selection and comparison will be discussed in more detail.

For each data set, an estimation procedure similar to the simulation study is performed for each of the four Rayleigh models. Thus, for a given model, MCMC is used to generate 10000 values from the posterior(s) of each model parameter(s), after which appropriate Bayesian estimates (corresponding to the four loss functions) are calculated. For the CRG model and its generalised counterpart, different priors are also taken into account. Finally, for every set of model parameter estimates across all models, a measure of goodness of fit known as the deviance information criterion (DIC) is calculated to allow for comparisons between results.

6.2 Model comparisons using the DIC

As stated above, the aim of this chapter is to illustrate the application of models and estimators derived in this thesis. It is thus of secondary importance to use some measure of fit in comparing the different models, given a specific data set. In practice, the Bayesian analyst should ideally choose a probability model, prior distribution and loss function based on external reasoning, and regard these choices as assumptions throughout the analysis. If these things are assumed, it does not make sense to calculate and compare results for a plethora of models and estimators. This exercise is performed here, however, out of interest to see how the different modelling procedures fare when quantified by a Bayesian measure of model fit, specifically the deviance information criterion (DIC).

Consider observed data $\mathbf{t}$ and a probability model with parameter(s) θ such that the likelihood is given by $\mathcal{L}(\theta|\mathbf{t})$ as described in Section 2.1.1. Arguably, the most popular model comparison measures are the Akaike information criterion (AIC) and the Bayesian information criterion (BIC), both of which are based on the deviance, defined as

$$D(\theta) = -2 \ln \mathcal{L}(\theta|\mathbf{t}). \tag{6.1}$$

A smaller deviance will generally indicate a better model, but needs to be penalised by the effective number of parameters, otherwise more complex models, which may overfit, will be favoured. Even though the AIC and BIC were formulated with completely different goals (the former favours models with good predictive ability, while the latter tries to identify the "true" probability generating mechanism), they are often very similar as to which models they select (Spiegelhalter et al., 2014). However, both of these

approaches use maximum likelihood estimates for θ in (6.1). The AIC and BIC differ in the way that the deviance is penalised as more model parameters are included.

Shortly after the widespread use of MCMC methods for Bayesian analyses began to take hold, Spiegelhalter et al. (2002) postulated a model comparison criterion, based loosely on the AIC, specifically for Bayesian settings. The posterior mean deviance,

$$\bar{D} = \mathrm{E}_{\theta|\mathbf{t}}[D(\theta)],$$

is suggested as a measure of goodness of fit, but the authors conceded that they were unsure how it should be penalised. In their original paper, the effective number of parameters was approximated by subtracting the deviance of posterior means from the posterior mean deviance, such that this penalty term becomes

$$p_D = \bar{D} - D(\hat{\theta}),$$

where $\hat{\theta}$ is the posterior mean of the parameter(s)–this amounts to using the Bayesian estimator under SE loss, but other loss functions may also be considered. The DIC can then be defined as

$$\mathrm{DIC} = \bar{D} + p_D. \tag{6.2}$$

The simplicity and ease with which the DIC could be calculated alongside MCMC procedures lead to its very extensive use in applied statistical analyses, but not without receiving a substantial string of criticisms from the Statistics community (Spiegelhalter et al., 2014). For example, Celeux et al. (2006) emphasise the weak theoretical justification of the DIC and conclude that it is not universally applicable. Delving deep into the statistical properties of the DIC is beyond the scope of this thesis, but two amendments to (6.2) that address some of the other criticisms are considered. Firstly, the effective number of parameters p_D has been shown to sometimes behave nonsensically (like taking on negative values) and moreover, is not invariant under reparameterisation (Spiegelhalter et al., 2014). An alternative form, p_V, proposed by Gelman et al. (2004),

$$p_V = \mathrm{V}_{\theta|\mathbf{t}}[D(\theta)], \tag{6.3}$$

the posterior variance of the deviance, leads to a more robust estimate of the number of effective parameters. By adjusting (6.2) accordingly, the amended form becomes

$$\mathrm{DIC}_V = \mathrm{DIC} + p_V. \tag{6.4}$$

Secondly, a number of authors have argued that the DIC favours more complex models, because it is not based on a proper predictive distribution (Ando, 2011). A modified version of (6.2), which is more heavily penalised by the effective number of parameters, is proposed to attenuate the tendency to overfit.

$$\mathrm{DIC}^* = \mathrm{DIC} + p_D = \bar{D} + 2p_D. \tag{6.5}$$

For the two example data sets that follow, the mended versions of the DIC, (6.4) and (6.5), will be investigated for each model. Two sets of these values are calculated, corresponding to estimates of model parameters under the symmetric loss functions.

6.3 Wind speed data

A data set consisting of average daily wind speeds at Elanora Heights, a suburb in Sydney, Australia, during November 2007 (Best et al., 2010) is considered. This data is used as an example by Barot and Patel (2014) and following their convention, it is also transformed from kilometre/hour to metre/second in the application here. They show that the Rayleigh distribution provides a fit to the data, with a Kolmogorov-Smirnov test showing no significant difference between distributions for a parameter value of 0.58538, with test statistic 0.20475 and p-value 0.14012 (Barot and Patel, 2014). The Kolmogorov-Smirnov test is a non-parametric way to determine if two samples emerge from the same probability distribution, with a null hypothesis stating that the distributions are equivalent. Thus, their findings motivate the choice of using the compounded Rayleigh models to describe the data.

One thing to note, however, is that survival analysis techniques would usually not be used for the wind speed data set, as its nature is inherently different from lifetime observations. This leaves room to artificially apply a censoring scheme and compare results in the presence and absence of censored observations. The censoring was applied

in a similar random manner to what was discussed in Section 3.3.2, with δ chosen to be 0.8. The data set was reordered, such that the first 23 observations are non-censored and the remaining 7 right-censored. Table 6.1 shows this arrangement. The data is analysed twice, first as if no censoring is present and then with the level of censoring specified above.

Table 6.1: *Wind speed data (in m/s). Observations artificially censored marked with an asterisk.*

0.5833	0.6667	0.6944	0.7222	0.7500	0.7778	0.8056	0.8056	0.8889	0.9167
1.0000	1.1111	1.1111	1.1667	1.1667	1.1944	1.2778	1.2778	1.3056	1.3333
1.3611	2.1111	2.1389	0.8611*	1.0278*	1.0278*	1.1111*	1.3333*	1.4444*	2.7778*

The CRE distribution is the first model under consideration. Table 6.2 below shows the general results calculated from the posterior realisations. A separate subtable also contains the results of the Bayesian estimators $\hat{\gamma}_{\text{LNX(a)}}$ and $\hat{\gamma}_{\text{GE(k)}}$ for varying values of the asymmetry parameters a and k. In all cases, these values were chosen to correspond with what was observed in the previous chapters' simulation studies, but remain largely arbitrary for this application.

Table 6.2: *Results for wind data, with no censoring, using the CRE model. The bottom table shows Bayesian point estimates for varying values of the asymmetry parameters.*

parameter	prior	95% HPD interval	median ($\hat{\gamma}_{\text{AE}}$)	mean ($\hat{\gamma}_{\text{SE}}$)	SD
γ	Jeffreys	(0.6959, 1.9772)	1.1828	1.2255	0.3340

	1	2	5	10
$\hat{\gamma}_{\text{LNX(a)}}$	1.1743	1.1302	1.0257	0.9027
$\hat{\gamma}_{\text{GE(k)}}$	1.1400	1.0988	0.9801	0.8110

In Table 6.3, a similar set of results are shown for the censored case. The variance increased and the estimates appear inflated, suggesting a similar trend of overestimation as was seen in Section 4.1.

Table 6.3: *Results for wind data, with 20% censoring, using the CRE model. The bottom table shows Bayesian point estimates for varying values of the asymmetry parameters.*

parameter	prior	95% HPD interval	median ($\hat{\gamma}_{\text{AE}}$)	mean ($\hat{\gamma}_{\text{SE}}$)	SD
γ	Jeffreys	(0.8307, 2.6702)	1.4792	1.5526	0.4824

	1	2	5	10
$\hat{\gamma}_{\text{LNX(a)}}$	1.4511	1.3714	1.2047	1.0419
$\hat{\gamma}_{\text{GE(k)}}$	1.4163	1.3534	1.1878	0.9865

Table 6.4: *Results for wind data, with no censoring, using the CRG model. The bottom tables show Bayesian point estimates for varying values of the asymmetry parameters.*

parameter	prior	95% HPD interval	median	mean	SD
α	Jeffreys	(2.4903, 87.5217)	14.1301	23.6742	23.8938
	Reference	(2.9123, 98.3892)	31.5659	37.9505	29.4256
	PM	(2.0971, 79.9998)	9.2758	18.5368	21.3639
β	Jeffreys	(3.1545, 132.1157)	21.3792	36.1123	36.2756
	Reference	(3.5277, 141.2125)	48.098	58.1404	44.5776
	PM	(2.4807, 123.9917)	13.6933	28.424	33.4594

		-5	-1	1	5	10
$\hat{\alpha}_{\text{LNX(a)}}$	Jeffreys	∞	136.3995	4.8848	2.5489	2.0104
	Reference	125.9663	119.2089	5.3697	2.5380	1.9407
	PM	116.9794	110.4445	4.2682	2.3803	1.8853
$\hat{\alpha}_{\text{GE(k)}}$	Jeffreys	54.8590	23.6742	9.2155	3.7627	2.5216
	Reference	64.7561	37.9505	13.4778	3.9635	2.4419
	PM	49.0157	18.5368	6.8524	3.2562	2.2817

		0.5	1	2	5
$\hat{\beta}_{\text{LNX(a)}}$	Jeffreys	7.6695	5.6004	4.0711	2.7288
	Reference	8.5471	5.9044	4.0831	2.6620
	PM	6.4828	4.9007	3.6786	2.5054
$\hat{\beta}_{\text{GE(k)}}$	Jeffreys	16.4259	12.7761	8.5879	4.4133
	Reference	26.3014	18.4891	10.3977	4.3849
	PM	11.6359	9.2015	6.5432	3.7258

Tables 6.4 and 6.5 shows the results when the CRG model with each of the three prior distributions was applied. In accordance with the findings in Section 4.2, α seems as if it may be underestimated when censoring is applied. Many of the estimates for β are decreased as well, but censoring does not seem to have a large effect. Interestingly, standard errors are affected very little, except those corresponding with the reference prior, where the censored cases counter-intuitively show a reduction.

The results of the analysis with the GCRE model can be found in Tables 6.6 and 6.7. The model parameter estimates reflect a slight increase in α-values and a decrease in c as censoring is applied. The variances are also slightly higher.

Lastly, the versatile GCRG distribution is considered as a model for the wind speed data. Tables 6.8 and 6.9 show the results for the non-censored and censored cases respectively. Considering the Bayesian estimates of the model parameters, decreases in estimates of both α and β can be seen as censoring is applied, while those of c remain largely unchanged. In almost all cases, it appears that the standard deviations of the estimates

Table 6.5: *Results for wind data, with 20% censoring, using the CRG model. The bottom tables show Bayesian point estimates for varying values of the asymmetry parameters.*

parameter	prior	95% HPD interval	median	mean	SD
α	Jeffreys	(1.3268, 47.1976)	5.6255	10.4034	12.0589
	Reference	(1.4471, 76.7342)	10.8498	21.1702	22.5806
	PM	(0.8969, 52.3457)	3.8417	9.5602	13.5851
β	Jeffreys	(1.8486, 101.0908)	10.6665	20.7405	24.7008
	Reference	(2.1489, 138.5652)	21.5217	40.8513	42.4291
	PM	(1.2178, 106.2497)	6.7902	18.8573	27.7924

		-5	-1	1	5	10
$\hat{\alpha}_{\text{LNX(a)}}$	Jeffreys	90.5212	83.2038	3.1866	1.6972	1.2499
	Reference	103.9197	96.6604	3.6355	1.7225	1.2156
	PM	82.4233	75.7032	2.5247	1.3391	0.9755
$\hat{\alpha}_{\text{GE(k)}}$	Jeffreys	31.7707	10.4034	4.2635	1.8782	1.1641
	Reference	48.3293	21.1702	5.9768	1.8782	1.0369
	PM	32.2098	9.5602	2.9657	1.2182	0.7561

		0.5	1	2	5
$\hat{\beta}_{\text{LNX(a)}}$	Jeffreys	5.6293	4.1776	3.0437	1.9263
	Reference	6.4853	4.5999	3.2126	1.9842
	PM	4.3786	3.2813	2.4096	1.5650
$\hat{\beta}_{\text{GE(k)}}$	Jeffreys	9.0680	7.1663	4.9205	2.3857
	Reference	14.2236	10.0440	5.9783	2.5758
	PM	6.1070	4.7104	3.2062	1.5941

Table 6.6: *Results for wind data, with no censoring, using the GCRE model. The bottom tables show Bayesian point estimates for varying values of the asymmetry parameters.*

parameter	prior	95% HPD interval	median	mean	SD
γ	Jeffreys	(0.7139, 2.5607)	1.3749	1.4326	0.4783
c	Jeffreys	(3.7426, 6.7982)	5.1467	5.1815	0.7813

	-2	-1	1	2	5	10
$\hat{\gamma}_{\text{LNX(a)}}$	1.7859	1.5707	1.3335	1.2559	1.0903	0.9163
$\hat{\gamma}_{\text{GE(k)}}$	1.5103	1.4326	1.2874	1.2188	1.0237	0.7690

	-5	-2	-1	1	2	5
$\hat{c}_{\text{LNX(a)}}$	6.8537	5.8451	5.5030	4.8959	4.6501	4.1118
$\hat{c}_{\text{GE(k)}}$	5.4119	5.2400	5.1815	5.0629	5.0031	4.8229

corresponding with the PM prior are less than those corresponding to the Jeffreys and reference priors, which were very similar.

Tables 6.10 and 6.11 show the adjusted DIC scores calculated across the models' parameter estimates for the non-censored and censored cases respectively. It is interesting

Table 6.7: *Results for wind data, with 20% censoring, using the GCRE model. The bottom tables show Bayesian point estimates for varying values of the asymmetry parameters.*

parameter	prior	95% HPD interval	median	mean	SD
α	Jeffreys	(0.9111, 3.2994)	1.7154	1.8213	0.6173
c	Jeffreys	(2.9888, 5.8448)	4.2515	4.2979	0.7323

	-2	-1	1	2	5	10
$\hat{\gamma}_{\text{LNX(a)}}$	1.7859	1.5707	1.3335	1.2559	1.0903	0.9163
$\hat{\gamma}_{\text{GE(k)}}$	1.5103	1.4326	1.2874	1.2188	1.0237	0.7690

	-5	-2	-1	1	2	5
$\hat{c}_{\text{LNX(a)}}$	6.8537	5.8451	5.5030	4.8959	4.6501	4.1118
$\hat{c}_{\text{GE(k)}}$	5.4119	5.2400	5.1815	5.0629	5.0031	4.8229

to note that the generalised models have much lower scores, suggesting better model fit. Between the non-generalised models, the CRE model seemed to do better than the CRG. Among the models for which an array of prior distributions were considered, the PM prior attains the best model fit. Overall, the CRG model with PM prior, using Bayesian estimates under SE loss yielded the lowest DIC scores.

For all models, estimates of the hazard rate were also obtained. Figure 6.1 shows the results for the CRE model and its generalised counterpart, the GCRE model, while Figures 6.2 and 6.3 show the results for the CRG and GCRG models respectively. The forms of the hazard rate for all models are relatively similar, except in the case of the CRG model, where the reference prior lead to outlying forms. In most cases, there was only a small discrepancy between estimates obtained using different loss functions. Arbitrary hyper-parameter values were chosen for the asymmetric loss functions for illustrative purposes.

Table 6.8: *Results for wind data, with no censoring, using the GCRG model. The bottom tables show Bayesian point estimates for varying values of the asymmetry parameters.*

parameter	prior	95% HPD interval	median	mean	SD
α	Jeffreys	(0.1989, 2.2763)	0.5479	0.7019	0.5897
	Reference	(0.2143, 2.0142)	0.4949	0.6474	0.4806
	PM	(0.1632, 1.5113)	0.431	0.5288	0.357
β	Jeffreys	(0.0351, 4.4514)	0.5184	0.9027	1.2408
	Reference	(0.039, 3.8708)	0.4478	0.8194	1.034
	PM	(0.013, 2.6747)	0.3211	0.5883	0.7437
c	Jeffreys	(3.4755, 12.29)	6.6629	6.9738	2.2005
	Reference	(3.7308, 12.297)	6.9834	7.2025	2.2038
	PM	(4.0399, 14.2176)	7.4119	7.9636	2.709

		-5	-2	-1	1	2	5
$\hat{\alpha}_{\text{LNX(a)}}$	Jeffreys	4.9002	2.7776	1.2071	0.5985	0.5478	0.4652
	Reference	3.3114	1.5304	0.8556	0.5659	0.5189	0.4428
	PM	1.6315	0.7822	0.6152	0.4780	0.4433	0.3800
$\hat{\alpha}_{\text{GE(k)}}$	Jeffreys	1.9118	0.9167	0.7019	0.4884	0.4276	0.3132
	Reference	1.4552	0.8063	0.6474	0.4618	0.4098	0.3166
	PM	1.0374	0.6380	0.5288	0.3784	0.3297	0.2356

		0.5	1	2	5
$\hat{\beta}_{\text{LNX(a)}}$	Jeffreys	0.6843	0.5854	0.4751	0.3297
	Reference	0.6402	0.5457	0.4402	0.3066
	PM	0.4845	0.4204	0.3417	0.2348
$\hat{\beta}_{\text{GE(k)}}$	Jeffreys	0.3232	0.1937	0.0591	0.0106
	Reference	0.3009	0.1993	0.0733	0.0078
	PM	0.1509	0.0596	0.0088	0.0013

		1	2	5	10	15
$\hat{c}_{\text{LNX(a)}}$	Jeffreys	5.4324	4.6179	3.4775	2.8386	2.5984
	Reference	5.6060	4.8297	3.7784	3.1826	2.9516
	PM	5.9805	5.1121	4.0069	3.4367	3.2126
$\hat{c}_{\text{GE(k)}}$	Jeffreys	6.3329	6.0249	5.1517	4.0199	3.3753
	Reference	6.5557	6.2465	5.4197	4.4330	3.8421
	PM	7.1531	6.7912	5.8620	4.7818	4.1723

Table 6.9: *Results for wind data, with 20% censoring, using the GCRG model. The bottom tables show Bayesian point estimates for varying values of the asymmetry parameters.*

parameter	prior	95% HPD interval	median	mean	SD
α	Jeffreys	(0.1261, 1.4277)	0.3322	0.4351	0.3418
	Reference	(0.1299, 2.3465)	0.389	0.5692	0.6463
	PM	(0.1264, 0.9776)	0.2868	0.3531	0.2565
β	Jeffreys	(0.0084, 3.1692)	0.279	0.5919	0.8678
	Reference	(0.0185, 5.1563)	0.4007	0.8994	1.556
	PM	(0.0096, 1.9415)	0.2012	0.3968	0.5955
c	Jeffreys	(3.4428, 13.7053)	6.8962	7.3636	2.7076
	Reference	(3.0128, 12.4353)	6.3129	6.6723	2.4014
	PM	(3.9457, 13.6199)	7.3929	7.7831	2.5455

		-5	-2	-1	1	2	5
$\hat{\alpha}_{\text{LNX(a)}}$	Jeffreys	2.0084	0.7254	0.5193	0.3897	0.3600	0.3075
	Reference	6.3999	4.1390	1.6185	0.4618	0.4166	0.3479
	PM	1.6506	0.5399	0.4026	0.3277	0.3106	0.2778
$\hat{\alpha}_{\text{GE(k)}}$	Jeffreys	1.0098	0.5533	0.4351	0.2912	0.2518	0.1859
	Reference	2.2116	0.8612	0.5692	0.3394	0.2879	0.2053
	PM	0.8799	0.4364	0.3531	0.2609	0.2322	0.1676

		0.5	1	2	5
$\hat{\beta}_{\text{LNX(a)}}$	Jeffreys	0.4634	0.3936	0.3142	0.2122
	Reference	0.6170	0.5109	0.4012	0.2686
	PM	0.3336	0.2963	0.2497	0.1822
$\hat{\beta}_{\text{GE(k)}}$	Jeffreys	0.1275	0.0637	0.0194	0.0040
	Reference	0.2161	0.1150	0.0325	0.0062
	PM	0.1080	0.0575	0.0158	0.0031

		1	2	5	10	15
$\hat{c}_{\text{LNX(a)}}$	Jeffreys	5.3702	4.5330	3.4567	2.8295	2.5663
	Reference	4.9348	4.1516	3.0773	2.3790	2.1049
	PM	5.8250	4.9085	3.6539	2.9136	2.6240
$\hat{c}_{\text{GE(k)}}$	Jeffreys	6.4732	6.0722	5.0613	3.9609	3.3598
	Reference	5.8619	5.4820	4.4928	3.3281	2.6778
	PM	6.9969	6.6252	5.6164	4.3357	3.5746

Table 6.10: *Summary of DIC measures across all models, in the case of no censoring, for wind speed data.*

model	prior	parameter estimates	DIC^*	DIC_V
CRE	Jeffreys	$\hat{\gamma}_{AE}$	56.9626	56.9468
		$\hat{\gamma}_{SE}$	57.1857	56.8352
CRG	Jeffreys	$(\hat{\alpha}_{AE}, \hat{\beta}_{AE})$	70.8131	107.7949
		$(\hat{\alpha}_{SE}, \hat{\beta}_{SE})$	64.3679	111.0176
	Reference	$(\hat{\alpha}_{AE}, \hat{\beta}_{AE})$	64.0965	89.6865
		$(\hat{\alpha}_{SE}, \hat{\beta}_{SE})$	62.5918	90.4388
	PM	$(\hat{\alpha}_{AE}, \hat{\beta}_{AE})$	70.0927	105.3072
		$(\hat{\alpha}_{SE}, \hat{\beta}_{SE})$	61.7397	109.4837
GCRG	Jeffreys	$(\hat{\gamma}_{AE}, \hat{c}_{AE})$	19.1946	24.1316
		$(\hat{\gamma}_{SE}, \hat{c}_{SE})$	19.1978	24.1300
GCRG	Jeffreys	$(\hat{\alpha}_{AE}, \hat{\beta}_{AE}, \hat{c}_{AE})$	16.9612	40.5769
		$(\hat{\alpha}_{SE}, \hat{\beta}_{SE}, \hat{c}_{SE})$	6.6559	45.7296
	Reference	$(\hat{\alpha}_{AE}, \hat{\beta}_{AE}, \hat{c}_{AE})$	14.4651	35.6311
		$(\hat{\alpha}_{SE}, \hat{\beta}_{SE}, \hat{c}_{SE})$	3.9398	40.8938
	PM	$(\hat{\alpha}_{AE}, \hat{\beta}_{AE}, \hat{c}_{AE})$	13.1229	40.6670
		$(\hat{\alpha}_{SE}, \hat{\beta}_{SE}, \hat{c}_{SE})$	2.1467	46.1551

Table 6.11: *Summary of DIC measures across all models, in the case of 20% censoring, for wind speed data.*

model	prior	parameter estimates	DIC^*	DIC_V
CRE	Jeffreys	$\hat{\gamma}_{AE}$	57.4095	56.6103
		$\hat{\gamma}_{SE}$	57.1849	56.7226
CRG	Jeffreys	$(\hat{\alpha}_{AE}, \hat{\beta}_{AE})$	67.3077	85.9008
		$(\hat{\alpha}_{SE}, \hat{\beta}_{SE})$	60.5014	89.3040
	Reference	$(\hat{\alpha}_{AE}, \hat{\beta}_{AE})$	63.1725	79.9996
		$(\hat{\alpha}_{SE}, \hat{\beta}_{SE})$	57.9415	82.6151
	PM	$(\hat{\alpha}_{AE}, \hat{\beta}_{AE})$	67.5708	88.9588
		$(\hat{\alpha}_{SE}, \hat{\beta}_{SE})$	57.4592	94.0146
GCRE	Jeffreys	$(\hat{\gamma}_{AE}, \hat{c}_{AE})$	17.6982	16.5214
		$(\hat{\gamma}_{SE}, \hat{c}_{SE})$	17.3715	16.6847
GCRG	Jeffreys	$(\hat{\alpha}_{AE}, \hat{\beta}_{AE}, \hat{c}_{AE})$	12.5524	29.4489
		$(\hat{\alpha}_{SE}, \hat{\beta}_{SE}, \hat{c}_{SE})$	4.3817	33.5342
	Reference	$(\hat{\alpha}_{AE}, \hat{\beta}_{AE}, \hat{c}_{AE})$	11.9679	32.0699
		$(\hat{\alpha}_{SE}, \hat{\beta}_{SE}, \hat{c}_{SE})$	2.4223	36.8428
	PM	$(\hat{\alpha}_{AE}, \hat{\beta}_{AE}, \hat{c}_{AE})$	9.5604	24.3053
		$(\hat{\alpha}_{SE}, \hat{\beta}_{SE}, \hat{c}_{SE})$	2.7436	27.7137

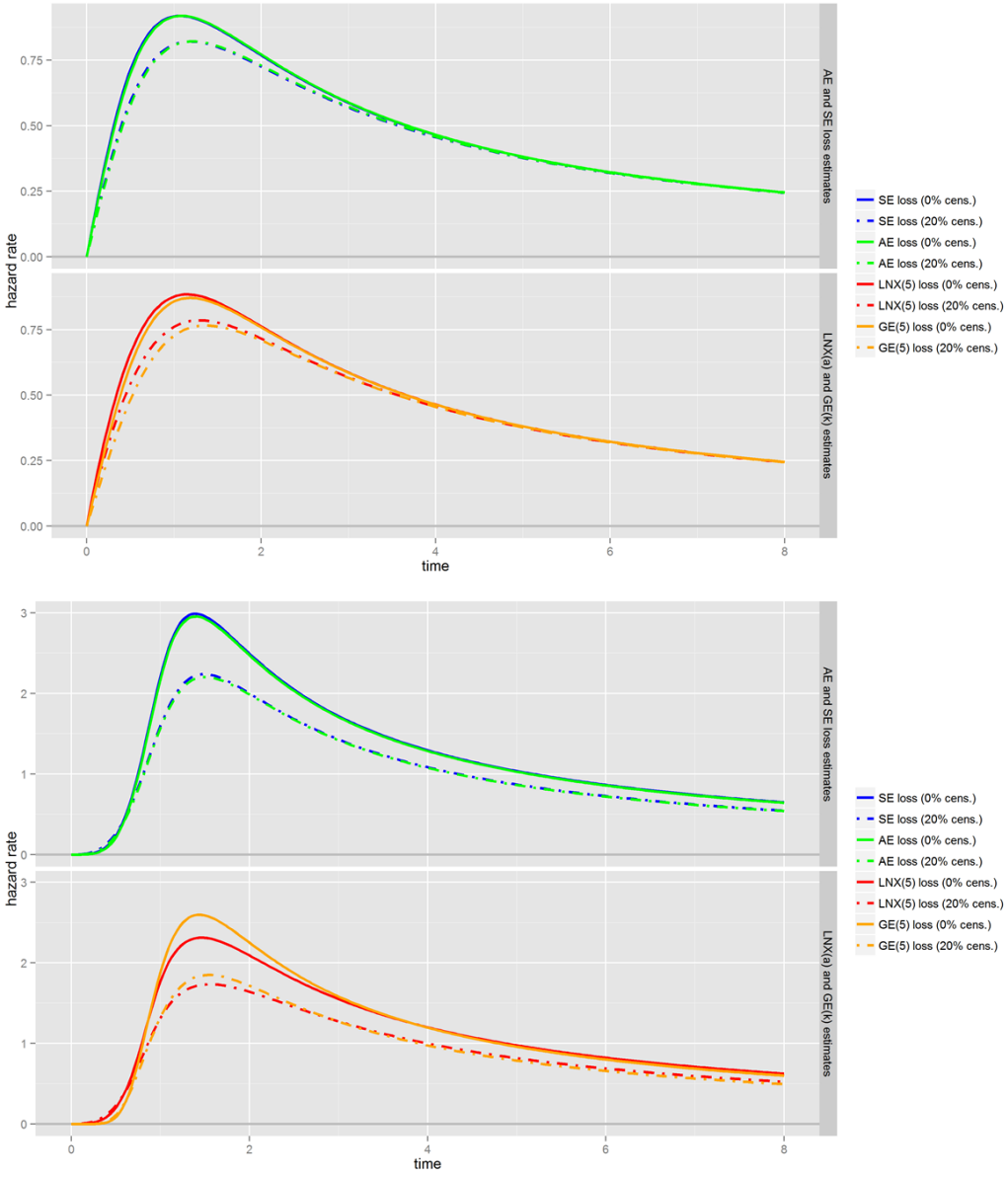


Figure 6.1: *Plots of Bayesian estimates of the hazard function for the wind speed data, using the CRE model (top) and GCRE model (bottom).*

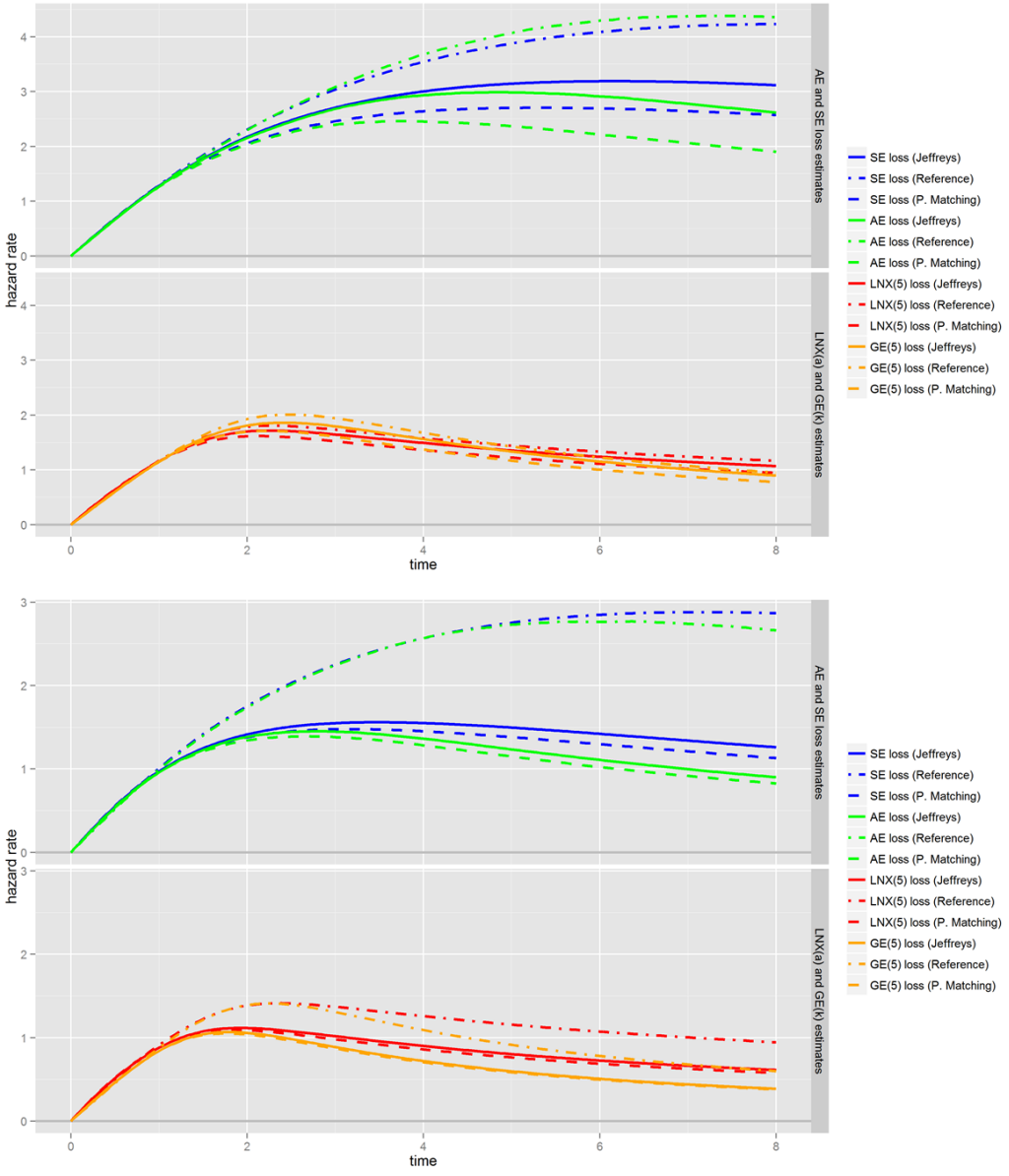


Figure 6.2: *Plots of Bayesian estimates of the hazard function for the wind speed data, using the CRG model with no censoring (top) and 20% censoring (bottom).*

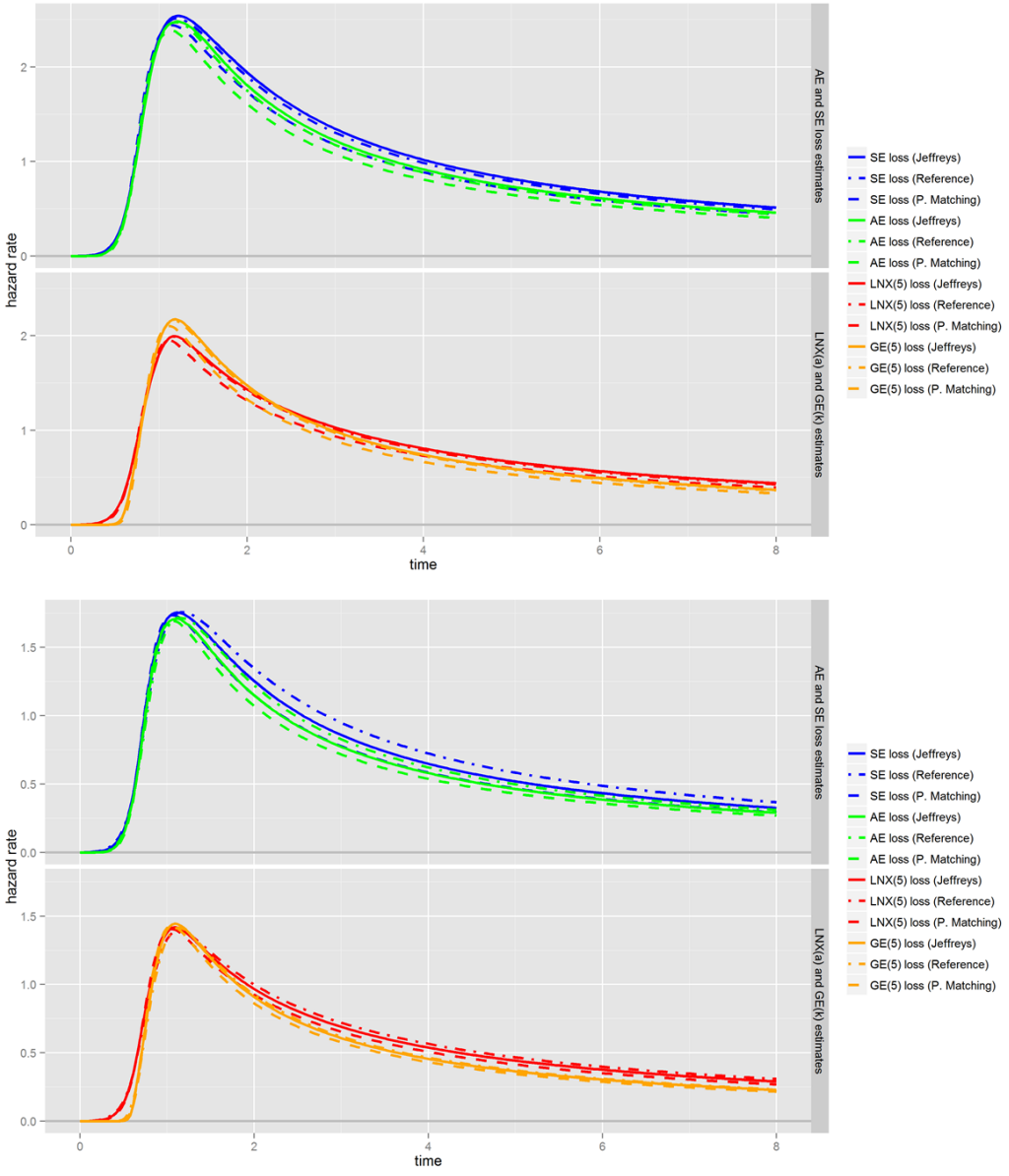


Figure 6.3: *Plots of Bayesian estimates of the hazard function for the wind speed data, using the GCRG model with no censoring (top) and 20% censoring (bottom).*

6.4 Gastrointestinal cancer data

To further illustrate the performance of the models and estimators derived in this study, the survival times of gastrointestinal cancer patients who received chemotherapy and radiation treatment are considered. The 45 observations, of which 8 are right-censored, was originally reported by Stablein et al. (1981), but analysed using various compound Rayleigh models by Bekker et al. (2000) and Abushal (2011).

Table 6.12: *Survival times of gastrointestinal cancer patients (in years). Censored observations are marked with an asterisk.*

0.047	0.115	0.121	0.132	0.164	0.197	0.203	0.260	0.282	0.296
0.334	0.395	0.458	0.466	0.501	0.507	0.529	0.534	0.540	0.570
0.641	0.644	0.696	0.841	0.863	1.099	1.219	1.271	1.326	1.447
1.485	1.553	1.581	1.589	2.178	2.343	3.743	2.416*	2.444*	2.825*
2.830*	3.578*	3.658*	3.978*	4.033*					

The data in Table 6.12 are modelled with the four compound Rayleigh distributions. For each, Bayesian estimates corresponding to all loss functions are calculated. The results for the CRE model are summarised in Table 6.13. Previously, it was observed that censoring caused overestimation of the true parameter value. This is a notable trend to take into account when interpreting the findings.

Table 6.13: *Results for cancer data using the CRE model. The bottom table shows Bayesian point estimates for varying values of the asymmetry parameters.*

parameter	prior	95% HPD interval	median ($\hat{\gamma}_{\text{AE}}$)	mean ($\hat{\gamma}_{\text{SE}}$)	SD
γ	Jeffreys	(0.3862, 1.1585)	0.6551	0.6927	0.2049

	1	2	5	10
$\hat{\gamma}_{\text{LNX(a)}}$	0.6730	0.6555	0.6127	0.5609
$\hat{\gamma}_{\text{GE(k)}}$	0.6382	0.6133	0.5461	0.4570

Table 6.14 shows the results when modelling the cancer data with the CRG. The estimates obtained with different prior distributions are very similar, except that the parameter estimates (and standard errors) corresponding to the PM prior are slightly lower.

Turning the attention to generalised models, the results of the GCRE and GCRG models are given in Tables 6.15 and 6.16 respectively. Both sets of results seem to suggest that the generalisation parameter c is between 1 and 2. For the generalised GCRG model, the PM prior leads to estimates with the lowest standard errors all around, followed by

Table 6.14: *Results for cancer data using the CRG model. The bottom tables show Bayesian point estimates for varying values of the asymmetry parameters.*

parameter	prior	95% HPD interval	median	mean	SD
α	Jeffreys	(0.2444, 0.6172)	0.3893	0.3993	0.1007
	Reference	(0.2494, 0.6214)	0.3878	0.3988	0.0984
	PM	(0.2511, 0.6442)	0.3818	0.397	0.0992
β	Jeffreys	(0.0369, 0.3469)	0.1209	0.1397	0.0815
	Reference	(0.0413, 0.3212)	0.1242	0.1392	0.0766
	PM	(0.0386, 0.3523)	0.1231	0.1416	0.0804

		-5	-1	1	5	10
$\hat{\alpha}_{\text{LNX(a)}}$	Jeffreys	0.4283	0.4045	0.3944	0.3765	0.3574
	Reference	0.4271	0.4038	0.3941	0.3773	0.3600
	PM	0.4267	0.4021	0.3923	0.3757	0.3592
$\hat{\alpha}_{\text{GE(k)}}$	Jeffreys	0.4509	0.3993	0.3749	0.3278	0.2761
	Reference	0.4497	0.3988	0.3763	0.3376	0.3001
	PM	0.4505	0.3970	0.3748	0.3368	0.2975

		0.5	1	2	5
$\hat{\beta}_{\text{LNX(a)}}$	Jeffreys	0.1381	0.1365	0.1336	0.1262
	Reference	0.1378	0.1364	0.1338	0.1271
	PM	0.1400	0.1385	0.1356	0.1282
$\hat{\beta}_{\text{GE(k)}}$	Jeffreys	0.1106	0.1014	0.0816	0.0280
	Reference	0.1137	0.1060	0.0920	0.0577
	PM	0.1133	0.1048	0.0895	0.0570

those corresponding to the Jeffreys prior. It is also interesting to note that for all priors, the estimates for c are fairly close to 2, leading to distributions similar in form to the CRG model.

Table 6.15: *Results for cancer data using the GCRE model. The bottom tables show Bayesian point estimates for varying values of the asymmetry parameters.*

parameter	prior	95% HPD interval	median	mean	SD
γ	Jeffreys	(0.4558, 1.3073)	0.7774	0.8008	0.2239
c	Jeffreys	(1.0526, 1.7261)	1.3500	1.3595	0.1715

	-2	-1	1	2	5	10
$\hat{\gamma}_{\text{LNX(a)}}$	0.8571	0.8273	0.7770	0.7555	0.7013	0.6323
$\hat{\gamma}_{\text{GE(k)}}$	0.8315	0.8008	0.7410	0.7117	0.6226	0.4881

	-5	-2	-1	1	2	5
$\hat{c}_{\text{LNX(a)}}$	1.4420	1.3903	1.3746	1.3452	1.3315	1.2941
$\hat{c}_{\text{GE(k)}}$	1.4032	1.3703	1.3595	1.3384	1.3280	1.2981

The summary of the adjusted DIC measures in Table 6.17 seem to suggest there was a payoff not only by generalising the compound Rayleigh models, but also in the addition

Table 6.16: *Results for cancer data using the GCRG model. The bottom tables show Bayesian point estimates for varying values of the asymmetry parameters.*

parameter	prior	95% HPD interval	median	mean	SD
α	Jeffreys	(0.2097, 1.7623)	0.4366	0.538	0.4412
	Reference	(0.2133, 1.1277)	0.4194	0.4832	0.2311
	PM	(0.1982, 1.0254)	0.3877	0.4456	0.2359
β	Jeffreys	(0.0193, 1.935)	0.1635	0.3131	0.5991
	Reference	(0.0157, 1.0073)	0.158	0.2416	0.2639
	PM	(0.0137, 0.9122)	0.1238	0.2044	0.2767
c	Jeffreys	(1.1085, 2.764)	1.8415	1.8731	0.4246
	Reference	(1.1867, 2.8462)	1.8506	1.9003	0.4192
	PM	(1.2285, 2.9496)	1.9308	1.9671	0.436

		-5	-2	-1	1	2	5
$\hat{\alpha}_{\text{LNX(a)}}$	Jeffreys	3.9734	2.0767	0.8319	0.4812	0.4544	0.4102
	Reference	0.8249	0.5578	0.5144	0.4598	0.4415	0.4033
	PM	1.2886	0.5657	0.4835	0.4230	0.4065	0.3726
$\hat{\alpha}_{\text{GE(k)}}$	Jeffreys	1.6131	0.6957	0.5380	0.4164	0.3844	0.3183
	Reference	0.7353	0.5356	0.4832	0.4063	0.3780	0.3133
	PM	0.8309	0.5042	0.4456	0.3714	0.3454	0.2901

		0.5	1	2	5
$\hat{\beta}_{\text{LNX(a)}}$	Jeffreys	0.2577	0.2318	0.2037	0.1642
	Reference	0.2260	0.2132	0.1936	0.1582
	PM	0.1889	0.1779	0.1621	0.1344
$\hat{\beta}_{\text{LNX(a)}}$	Jeffreys	0.1201	0.0866	0.0393	0.0088
	Reference	0.1113	0.0802	0.0397	0.0106
	PM	0.0902	0.0667	0.0331	0.0054

		1	2	5	10	15
$\hat{c}_{\text{LNX(a)}}$	Jeffreys	1.7889	1.7148	1.5409	1.3687	1.2782
	Reference	1.8189	1.7488	1.5882	1.4259	1.3334
	PM	1.8786	1.8021	1.6275	1.4493	1.3460
$\hat{c}_{\text{LNX(a)}}$	Jeffreys	1.7781	1.7309	1.5955	1.4191	1.3123
	Reference	1.8109	1.7677	1.6464	1.4857	1.3784
	PM	1.8726	1.8264	1.6958	1.5183	1.3969

of model parameters. As with the wind data, the generalised models yield much lower DIC scores than the non-generalised ones. The difference in model fit between parameter estimates within the models are negligible.

These comparisons propose that the GCRG model is again the best contender regarding model fit, similar to the findings regarding the wind speed data. However, if the true value of the generalisation parameter for this model really is close to 2, as Table 6.16 suggest, the form of the underlying distribution can be specified just as well with the non-generalised model. In fact, the principle of parsimony would compel us to rather

Table 6.17: *Summary of DIC measures across all models for cancer data.*

model	prior	parameter estimates	DIC^*	DIC_V
CRE	Jeffreys	$\hat{\gamma}_{AE}$	125.6468	125.9767
		$\hat{\gamma}_{SE}$	125.3857	126.1073
CRG	Jeffreys	$(\hat{\alpha}_{AE}, \hat{\beta}_{AE})$	112.7905	117.2113
		$(\hat{\alpha}_{SE}, \hat{\beta}_{SE})$	111.7656	117.7238
	Reference	$(\hat{\alpha}_{AE}, \hat{\beta}_{AE})$	110.1504	114.2976
		$(\hat{\alpha}_{SE}, \hat{\beta}_{SE})$	109.2903	114.7276
	PM	$(\hat{\alpha}_{AE}, \hat{\beta}_{AE})$	110.8619	116.5810
		$(\hat{\alpha}_{SE}, \hat{\beta}_{SE})$	109.7321	117.1459
GCRE	Jeffreys	$(\hat{\gamma}_{AE}, \hat{c}_{AE})$	67.4785	68.1032
		$(\hat{\gamma}_{SE}, \hat{c}_{SE})$	67.2463	68.2193
GCRG	Jeffreys	$(\hat{\alpha}_{AE}, \hat{\beta}_{AE}, \hat{c}_{AE})$	64.2571	80.9247
		$(\hat{\alpha}_{SE}, \hat{\beta}_{SE}, \hat{c}_{SE})$	56.5007	84.8029
	Reference	$(\hat{\alpha}_{AE}, \hat{\beta}_{AE}, \hat{c}_{AE})$	61.2530	73.9011
		$(\hat{\alpha}_{SE}, \hat{\beta}_{SE}, \hat{c}_{SE})$	56.4025	76.3264
	PM	$(\hat{\alpha}_{AE}, \hat{\beta}_{AE}, \hat{c}_{AE})$	61.7501	74.8221
		$(\hat{\alpha}_{SE}, \hat{\beta}_{SE}, \hat{c}_{SE})$	56.3592	77.5176

use the simpler model. The goodness of fit measures for this model is twice that of the generalised counterpart, though, suggesting that either the slight change in c is quite significant, or that the DIC is flawed in the current case.

Finally, Bayesian estimates for the hazard rates were also calculated. It is reassuring to note little difference between the forms of hazards rates corresponding to the CRE and GCRE models in Figure 6.4, and the CRG and GCRG models in Figure 6.5. It is even harder to distinguish between estimates obtained using different loss functions. Overall, the hazard rate seemed to be characterised by a very sharp initial increase.

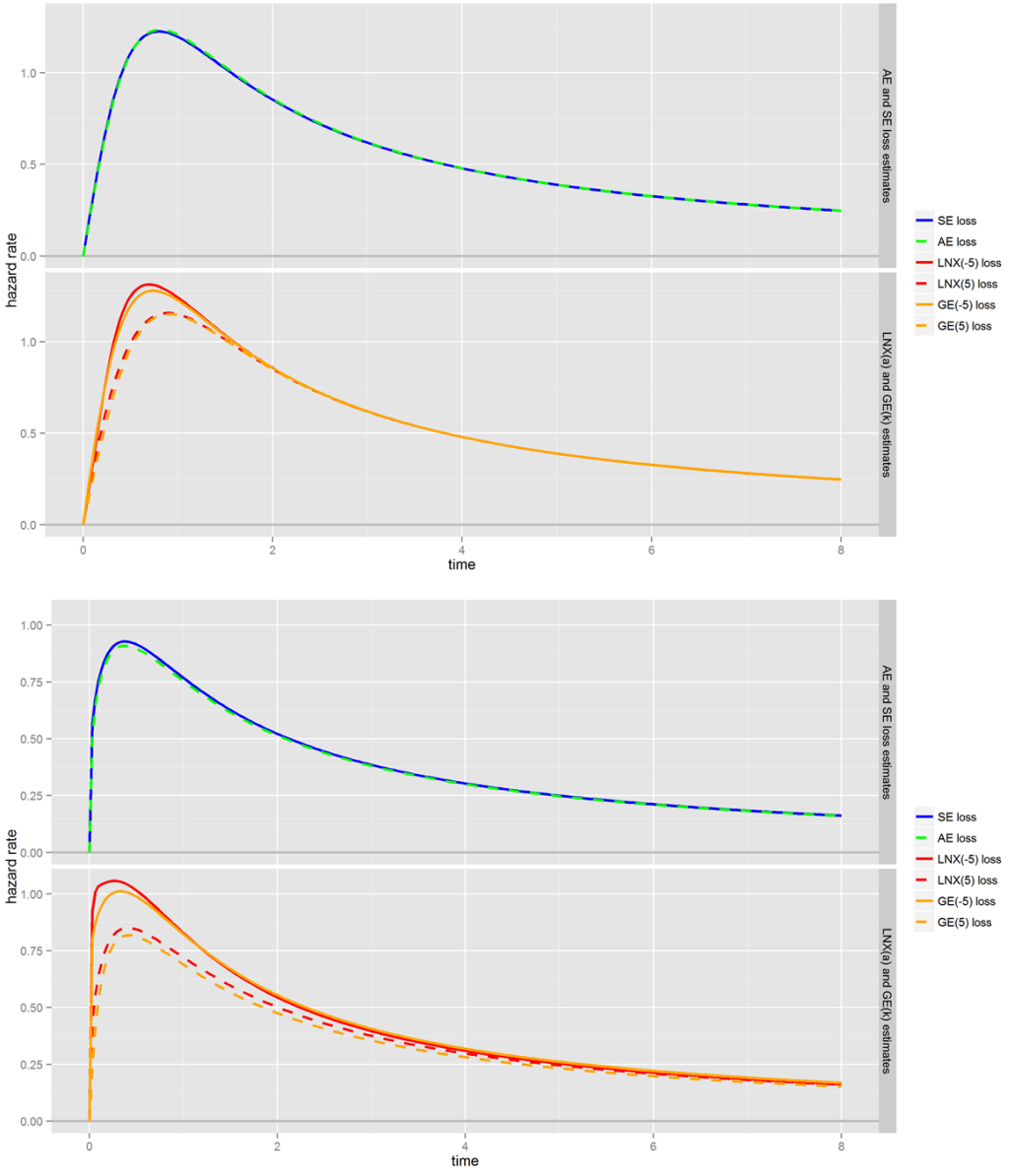


Figure 6.4: *Plots of Bayesian estimates of the hazard function for the cancer data, using the CRE model (top) and GCRE model (bottom).*

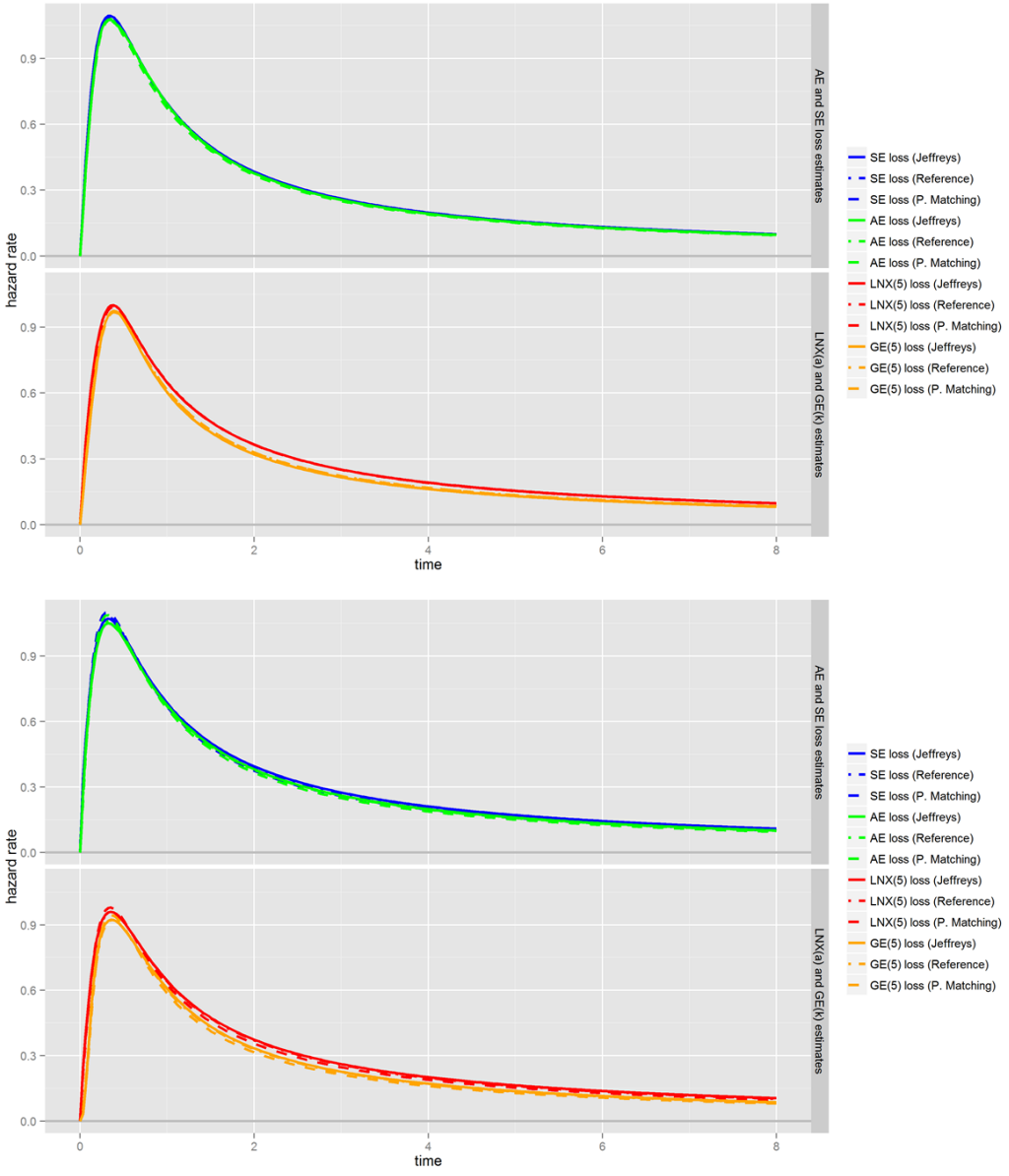


Figure 6.5: *Plots of Bayesian estimates of the hazard function for the cancer data, using the CRG model (top) and GCRG model (bottom).*

Chapter 7

Conclusions

In Chapters 4, 5 and 6, results of the simulation studies and application to real data were discussed. In this chapter, those results are summarised and trends that transcended across all compound Rayleigh models' simulations are outlined. Finally, in Section 7.5, recommendations for further work concerning this project that could potentially be done, are listed.

7.1 The effect of censoring on estimators

In the simulation studies and the application of the models to the wind speed data, it became clear that censoring has a very significant effect on the performance of the estimators. When comparing the accuracy and precision of Bayesian estimators in the cases of no censoring versus a moderate level of censoring (20%), it was observed that censoring not only caused the variances to increase, but introduced a degree of bias as well. The direction and magnitude of this bias seemed to differ between models and is dependent on the true value of the model parameters.

The increase in absolute bias could be neutralised by the asymmetric loss functions to some extent. By choosing the value of the asymmetry parameters a and k appropriately when respectively using the LINEX and GE loss functions, it was found that estimates very close to the true values could be obtained, despite the presence of censored observations. However, it is important to note that conceptually, the aim is not to optimise

the hyper-parameters of the loss functions and the choice of a and k should not rely on producing accurate estimators, but rather be based on external reasoning.

7.2 Comparison of loss functions

Loss functions play a central role in Bayesian estimation. Similar to the prior distribution, the loss function should be chosen *a priori* by the analyst to reflect the nature of the loss in the case of a specific scenario. If it is important that large errors must be penalised more than small ones, SE loss would be a suitable choice. Likewise, if overestimation is more severe, an asymmetric loss function such as LINEX or GE loss with appropriate choice of the hyper-parameter should be employed.

The Bayesian estimators were derived here with two symmetric and two asymmetric loss functions. The former, AE and SE loss, could be compared directly, since their estimators were not dependent on hyper-parameters. In almost all cases, estimators derived under AE loss were shown to have point estimates closer to true values, as well as lower measures of error (MAE and MSE).

The calculation of the asymmetric LINEX and GE loss estimates and corresponding error measures were largely for illustration rather than comparison. The nature of these loss functions, as well as how they are influenced by varying values of their hyper-parameters, meant that it was not sensible to compare them with each other or with the symmetric loss functions. By tuning a or k, the error measures for the estimators of these loss functions could potentially be made arbitrarily small. One interesting discrepancy was observed regarding the way in which the estimates varied as a function of the hyper-parameters. Overall, it appeared that the change in estimated values under GE loss as k was varied had more linearity than the change in estimates under LINEX loss as a was varied. Even though this does not necessarily make GE loss superior, it does suggest that adjusting the symmetry parameter of GE loss can have a more predictable outcome of the estimator than for LINEX loss. Nonetheless, the use of asymmetric loss functions should be situation-dependant, but their flexibility can make them an attractive modelling choice in many situations. In both cases, appropriate choices of the hyper-parameters can result in a loss function that is very close to symmetric.

7.3 Comparison of prior distributions

Two of the models of interest, the CRG model and its generalised counterpart, had multiple alternatives of posterior distributions for its model parameters, depending on which prior distribution was used for its derivation. The Jeffreys prior, reference prior and PM prior were considered. It should once again be noted that the latter has an equivalent form to one of the reference priors.

Although no single prior could clearly be distinguished as having the best performance, it seemed that the PM prior often achieved superior results, if only by a small margin. Even though it is usually not recommended for multidimensional situations, Jeffreys' prior also performed well. The reference prior at times resulted in very high bias and MSE measures, depending on the parameter configurations considered. This supports the idea that references priors should be derived with all orderings of the parameter and compared with one another.

The Jeffreys prior performed especially well when estimating the generalisation parameter c for the GCRG model. This may be due to the conditional argument on which the prior formulation was based.

7.4 Comparison of compound Rayleigh models

In this thesis, four compound Rayleigh models were derived and evaluated during the course of simulation studies. The aim was to investigate and compare non-informative prior distributions and loss functions, not at all to compare different Rayleigh models. Such a comparison was only carried out when applying these models in Chapter 6, but purely in the sake of interest.

In both applications, the models' goodness of fit was measured by two variants of the DIC. The most striking result was that the information criterion definitively favoured the generalised models over the non-generalised ones. Even though the validity of the DIC can be contested, it is an interesting outcome and suggests that the additional flexibility introduced by the generalisation parameter can indeed improve the models' performance. The most important consideration for model selection, though, should stem from external consideration regarding the specific experiment to which it is applied.

The choice of model should be made *a priori* and regarded as an assumption throughout the analysis.

7.5 Shortcomings and further work

Due to the nature of this project as well as time constraints, some aspects of the study could not be investigated as deeply as the author intended. Furthermore, some potential areas of interest emerged from the work done here, which unfortunately fell beyond the scope of the project.

7.5.1 Shortcomings

A few obvious shortcomings remain to be addressed. The most prominent shortcoming is that the effect of sample size was only partially investigated. The main reason for not prioritising this factor was the tremendous amount of computational time required to redo every simulation study with different sample sizes. An additional consideration was the relatively predictable outcome of an increase in sample size. It is to be expected that a larger sample size will lead to estimators with increased accuracy and precision and this is exactly what was observed in the cases where sample size were indeed investigated.

Another aspect that could readily be extended in this study regards the application of the non-informative prior theory. The GCRE model was an extension of the single parameter CRE model; the additional parameter was not necessarily considered as a parameter of interest. Nonetheless, it would have been interesting to derive reference and PM priors for the two-parameter model, for comparative purposes. Similarly, prior derivation for the GCRG model could have been considered in more detail, such as deriving three-parameter reference and PM priors, although this would have increased the computational burden significantly.

Even though simulating a posterior is trivial concerning the computational time it takes, the sheer amount of simulations performed resulted in procedures that took in the magnitude of hours to execute. One shortcoming in this regard is that extra precautions were not taken to make the simulation study more efficient. The choice of a lower level programming language, such as C, could have yielded dramatic reductions to the

computational time. Other MCMC algorithms could also have been considered for this purpose, such as the Gibbs sampler, in which every candidate value is accepted.

7.5.2 Potential for future work

Apart from the shortcomings, some potential avenues of exploration were knowingly omitted. This was mainly in order to keep the study concise. An example is the scope of the choice of loss functions and prior distributions. For the purposes here, it was decided beforehand that the study will compare two symmetric and two asymmetric loss functions, as well as three types of non-informative prior distributions. There is room to investigate a wider range of options that fell beyond this scope. For example, it might be interesting to see how the properties of the posterior distributions change if other priors, such as conjugate priors, are used for their derivation. Similarly, other loss functions could be considered, such as the weighted LINEX loss.

Another interesting alternative to the way in which loss functions were investigated here, would be to specify different loss functions for different time periods in the analysis. For example, this could be a sensible approach in settings where favouring over- or underestimation is sensible at the start of a study, but a symmetric penalty is desired at a later stage. This could lead to so-called bathtub hazard curves, where the hazard shows an additional increase after the initial decrease.

The Bayesian school of thought was considered here and there was no interest to compare estimation procedures with similar outcomes from the frequentist paradigm. That being said, it would be interesting to see how results from frequentist and maximum likelihood theory differ from what has been obtained in this thesis.

Lastly, the work done here only scratches the surface of Bayesian model selection, an active field of research with many disagreements as to which procedures are appropriate. It would be interesting to study methods for assessing the validity of compounded Rayleigh models (or in a wider scope), even though the choice of model should ideally be regarded as an *a priori* assumption during the analysis.

Appendix A

Additional definitions

A.1 The inverse transform method

The inverse transform method is a simple way to generate a random sample from any distribution for which the CDF is explicitly known. Suppose a random variable T is distributed according to a distribution with CDF F, and a realisation t is desired. The inverse transform principle states that $F^{-1}(V)$ will be distributed according to F, where V is a $U(0,1)$ distributed random variable.

$$t = F^{-1}(u).$$

A.2 The Beta prime distribution

The Beta prime distribution is a continuous probability distribution which is defined for a random variable $x > 0$. Its form is defined as follows:

$$f(x) = \frac{x^{\alpha-1}(1+x)^{-(r+s)}}{B(r,s)},$$

where B is the Beta function, thus $r, s > 0$. This PDF is helpful when solving some of the complicated integrals for the derivation of the compound models.

More specifically, we can state that

$$\int_0^\infty \frac{x^{r-1}(1+x)^{-(r+s)}}{B(r,s)}dx = 1$$

and upon making the transformation of variables $x = \left(\frac{t}{q}\right)^p$, this becomes

$$\int_0^\infty \frac{p\left(\frac{t}{q}\right)^{rp-1}\left[1+\left(\frac{t}{q}\right)^p\right]^{-(r+s)}}{qB(r,s)}dt = 1,$$

such that we have

$$\int_0^\infty p\left(\frac{t}{q}\right)^{rp-1}\left[1+\left(\frac{t}{q}\right)^p\right]^{-(r+s)}dt = qB(r,s).$$

The relation above becomes useful when solving integrals where the integrand has an equivalent form.

Appendix B

Software code

This appendix contains the code used for performing the simulation studies, as well as plotting all figures and results. All of the software was coded with the R statistical computing platform (R Core Team, 2012).

B.1 R code for distributions, priors and posteriors

```
##########  CDFS                                  ######################
#########################################################################


pray <- function(x, theta)
  return( 1-exp(-theta*x^2) )

pcrayexp <- function(x, gamma)
  return( 1-(1 + (x^2)/gamma)^(-1) )

pcraygam <- function(x, alpha, beta)
  return( 1-(1 + (x^2)/beta)^(-alpha) )

pgcrayexp <- function(x, gamma, cc)
  return( 1-(1 + (x^cc)/gamma)^(-1) )

pgcraygam <- function(x, alpha, beta, cc)
  return( 1-(1 + (x^cc)/beta)^(-alpha) )



##########  RAYLEIGH COMPOUNDED WITH EXPONENTIAL ######################
#########################################################################

dcrayexp <- function(t, gamma)
{
  return( 2*gamma*t*((t^2 + gamma)^(-2)) )
}

```

```

log_post_fun_crayexp <- function(gamma, tvec, d=length(tvec))
{
  ## Proportional posterior function for compound Rayleigh Exp distribution
  ## with parameter gamma, for a vector of times 'tvec' of which first 'd' are noncensored

  tvec_gam <- gamma + tvec^2
  u <- sum(log( (tvec[1:d])/(tvec_gam[1:d]) ))
  Tl <- sum(log(tvec_gam))

  return( (length(tvec)-1)*log(gamma) + u - Tl )
}


gen_tvec_crayexp <- function(n, gamma, delta=1)
{
  ## Generates vector of n times from a compound Rayleigh Exp model
  ## with parameter gamma and noncensored proportion delta

  tt <- sqrt((gamma/runif(n)) - gamma)

  if(delta == 1)
    return(c(n, tt))
  else{
    uu <- runif(n)
    return(c(sum(uu < delta), tt[uu < delta], tt[uu >= delta]))
  }
}


cre_h <- function(gamma, t)
  return( (2*t)/(gamma + t^2) )

cre_s <- function(gamma, t)
  return( 1/(1 + (t^2)/gamma) )



##########  RAYLEIGH COMPOUNDED WITH GAMMA       ######################
#############################################################################

dcraygam <- function(t, a, b)
{
  return( 2 * t * a * (b^a) * (b + t^2)^(-a -1) )
}


log_post_fun_craygam <- function(a, b, tvec, d=length(tvec), prior_t="jeff")
{
  tvec_beta <- b + tvec^2
  u <- sum(log(tvec[1:d] / tvec_beta[1:d]))
  Tl <- sum(log(tvec_beta/b))

  pd <- switch(prior_t,
               "jeff" = 1 / (b * (a+1) * sqrt(a * (a+2))),
               "ref1" = 1 / (a * b),
               "ref2" = 1 / (a * b * (a+1)),
               "pmp"  = 1 / (a * b * (a+1)))

  return( log(pd) + d*log(a) + u - a*Tl )
}



```

```
gen_tvec_craygam <- function(n, a, b, delta=1)
{
  ## Generates vector of n times from a compound Rayleigh Gam model
  ## with parameter a and b and noncensored proportion delta

  tt <- sqrt(b*( (runif(n))^(-(1/a)) -1))

  if(delta == 1)
    return(c(n, tt))
  else{
    uu <- runif(n)
    return(c(sum(uu < delta), tt[uu < delta], tt[uu >= delta]))
  }
}


crg_h <- function(alpha, beta, t)
  return( (2*alpha*t)/(beta + t^2) )

crg_s <- function(alpha, beta, t)
  return( (1 + (t^2)/beta)^(-alpha) )



##########  GENERALISED                            ######################
##########  RAYLEIGH COMPOUNDED WITH EXPONENTIAL   ######################
#########################################################################


dgcrayexp <- function(t, gamma, cc=2)
{
  return( cc*gamma*(t^(cc-1))*((t^cc + gamma)^(-2)) )
}


gcre_h <- function(gamma, cc, t)
  return( (cc*(t^(cc-1)))/(gamma + t^cc) )


gcre_s <- function(gamma, cc, t)
  return( (1 + (t^cc)/gamma)^(-1) )


gen_tvec_gcrayexp <- function(n, gamma, cc=2, delta=1)
{
  ## Generates vector of n times from a compound Rayleigh Exp model
  ## with parameter gamma and noncensored proportion delta

  tt <- ((gamma/runif(n)) - gamma)^(1/cc)

  if(delta == 1)
    return(c(n, tt))
  else{
    uu <- runif(n)
    return(c(sum(uu < delta), tt[uu < delta], tt[uu >= delta]))
  }
}


numerical_gcre_jeffreys <- function(gam, cc)
{
  A1_integrand <- function(t, gam, cc)
    return( ((t^(2*cc-1))*log(t))/((t^cc + gam)^4) )
  A2_integrand <- function(t, gam, cc)
    return( ((t^(2*cc-1))*(log(t)^2))/((t^cc + gam)^4) )
```

```
160
161   A1_term <- integrate(A1_integrand, lower=0, upper=Inf, gam=gam, cc=cc)
162   A2_term <- integrate(A2_integrand, lower=0, upper=Inf, gam=gam, cc=cc)
163
164   A1_term <- 4*(cc^2)*(gam^2)*(A1_term[[1]]^2)
165   A2_term <- 2*cc*gam*A2_term[[1]]
166
167   val <- ((1/(3*gam^2))*(1/(cc^2) + A2_term) - A1_term)
168   if(val <= 0)
169     val <- 0.0000001
170
171   return( val )
172 }
173
174
175 log_post_fun_gcrayexp <- function(gamma, cc, tvec, d=length(tvec))
176 {
177   ## Proportional posterior function for compound Rayleigh Exp distribution
178   ## with parameter gamma, for a vector of times 'tvec' of which first 'd' are noncensored
179
180   tvec_gam <- gamma + tvec^cc
181   u <- sum(log( (tvec[1:d]^(cc-1))/(tvec_gam[1:d]) ))
182   Tl <- sum(log(tvec_gam))
183
184
185   jeff_pr <- numerical_gcre_jeffreys(gamma, cc)
186   jeff_pr <- 0.5*log(jeff_pr)
187
188   return( jeff_pr + d*log(cc) + length(tvec)*log(gamma) + u - Tl )
189 }
190
191
192
193 ##########  GENERALISED                          ######################
194 ##########  RAYLEIGH COMPOUNDED WITH GAMMA       ######################
195 ##########################################################################
196
197
198 dgcraygam <- function(t, a, b, cc)
199 {
200   return( cc * (t^(cc-1)) * a * (b^a) * (b + t^cc)^(-a -1) )
201 }
202
203
204 gen_tvec_gcraygam <- function(n, a, b, cc, delta=1)
205 {
206   ## Generates vector of n times from a compound Rayleigh Gam model
207   ## with parameter a and b and noncensored proportion delta
208
209   tt <- (b*( (runif(n))^(-(1/a)) -1))^(1/cc)
210
211   if(delta == 1)
212     return(c(n, tt))
213   else{
214     uu <- runif(n)
215     return(c(sum(uu < delta), tt[uu < delta], tt[uu >= delta]))
216   }
217 }
218
219
220 gcrg_h <- function(alpha, beta, cc, t)
221   return( (cc*alpha*(t^(cc-1)))/(beta + t^cc) )
222
223 gcrg_s <- function(alpha, beta, cc, t)
224   return( (1 + (t^cc)/beta)^(-alpha) )
```

```
225
226
227 numerical_gcrg_jeffreys_cond <- function(a, b, cc)
228 {
229   A_integrand <- function(t, alpha, beta, cc)
230     return( ((t^(2*cc-1))*(log(t)^2))/((t^cc + beta)^(alpha+3) ) )
231
232   A_term <- integrate(A_integrand, lower=0, upper=Inf, alpha=a, beta=b, cc=cc)[[1]]
233
234   val <- (1/cc^2) + (a+1)*cc*(b^(a+1))*A_term
235   if(val <= 0)
236     val <- 0.0000001
237
238   return( val )
239 }
240
241
242 log_post_fun_gcraygam_cond <- function(a, b, cc, tvec, d=length(tvec), prior_type="jeff")
243 {
244   tvec_beta <- b + tvec^cc
245   u <- sum(log( ((tvec[1:d])^(cc-1)) / tvec_beta[1:d]))
246   Tl <- sum(log(tvec_beta/b))
247
248   pd <- switch(prior_type,
249                "jeff" = 1 / (b * (a+1) * sqrt(a * (a+2))),
250                "ref1" = 1 / (a * b),
251                "ref2" = 1 / (a * b * (a+1)),
252                "pmp"  = 1 / (a * b * (a+1)))
253
254   jeff_cc_cond <- numerical_gcrg_jeffreys_cond(a, b, cc)
255   log_pd <- log(pd) + 0.5*log(jeff_cc_cond)
256
257   return( log_pd + d*log(cc*a) + u - a*Tl )
258 }
```

B.2 R code for Metropolis-Hastings algorithms

```
1 ######################################################################
2 ########                                                      ########
3 ########           MCMC METROPOLIS-HASTINGS SAMPLING          ########
4 ########                 FROM UNIVARIATE TARGET               ########
5 ########                                                      ########
6 ######################################################################
7
8
9 metrop_hast_gamma <- function(k, ffun, theta0 = 1, gbeta = 1, ...)
10 {
11   ## Function performing Metropolis-Hastings algorithm to generate
12   ## ffun is the log of the proportional posterior distribution
13   ## Markov Chain with stationary distribution ffun.
14   ## ffun must have variable as first argument.
15   ## Starting value is specified in theta0.
16   ## Assumption of gamma proposal distribution, with first parameter
17   ## to be updated and second (fixed) parameter specified in gbeta.
18   ## '...' specifies optional arguments to ffun
19
20   acceptfun <- function(Y, X){
21     # function which determines if Y is accepted
22     return( ffun(Y, ...) + dgamma(X, Y, gbeta, log=T) - ffun(X, ...) - dgamma(Y, X, gbeta, log=T) )
23   }
24
```

```
25
26    theta <- c(theta0, numeric(k))
27    for(i in 1:k){
28      Y <- rgamma(1, theta[i], gbeta)
29      jj <- acceptfun(Y, theta[i])
30      if(is.nan(jj)) jj <- -Inf  # adjust output for very small floats
31      TST <- log(runif(1)) <= jj
32
33      if(TST) theta[i+1] <- Y
34      else theta[i+1] <- theta[i]
35    }
36
37    return(theta)
38  }
39
40
41  metrop_hast_gamma2 <- function(k, ffun, theta0 = 1, gbeta = 1, ...)
42  {
43    ## see metrop_hast_gamma()
44    ## when rejecting new realisations from proposal distribution
45    ## the previous accepted realisation is not repeated in the
46    ## current index (i.e. while instead of for)
47
48    acceptfun <- function(Y, X){
49      # function which determines if Y is accepted
50      return( ffun(Y, ...) + dgamma(X, Y, gbeta, log=T) - ffun(X, ...) - dgamma(Y, X, gbeta, log=T) )
51    }
52    theta <- c(theta0, numeric(k))
53
54    i <- 1
55    while(i <= k){
56      Y <- rgamma(1, theta[i], gbeta)
57      jj <- acceptfun(Y, theta[i])
58      if(is.nan(jj)) jj <- -Inf  # adjust output for very small floats
59      TST <- log(runif(1)) <= jj
60
61      if(TST) {
62        i <- i+1
63        theta[i] <- Y
64      }
65    }
66
67    return(theta)
68  }
69
70
71
72  ######################################################################
73  ########                                                      ########
74  ########           MCMC METROPOLIS-HASTINGS SAMPLING          ########
75  ########                  FROM BIVARIATE TARGET               ########
76  ########                                                      ########
77  ######################################################################
78
79
80  bv_metrop_hast_gamma <- function(k, ffun, theta0 = c(1,1), gbeta = 1, ...)
81  {
82    ## Function performing Metropolis-Hastings algorithm to generate
83    ## Markov Chain with bivariate stationary distribution ffun.
84    ## ffun must have variable as first argument and is specified in log form.
85    ## Starting values is specified in theta0.
86    ## Assumption of gamma proposal distribution, with first parameter
87    ## to be updated and second (fixed) parameter specified in gbeta.
88    ## '...' specifies optional arguments to ffun
89    ## Output chain of length k+1 to account for initial values
```

```
90
91    accfun1 <- function(Y, X){
92      # function which determines if Y is accepted
93      # (for first parameter)
94      return( ffun(Y, X[2], ...) + dgamma(X[1], Y, gbeta, log=T) - ffun(X[1], X[2], ...) - dgamma(Y, X[1],
          gbeta, log=T) )
95    }
96    accfun2 <- function(Y, X){
97      # function which determines if Y is accepted
98      # (for second parameter)
99      return( ffun(X[1], Y, ...) + dgamma(X[2], Y, gbeta, log=T) - ffun(X[1], X[2], ...) - dgamma(Y, X[2],
          gbeta, log=T) )
100   }
101
102   theta <- rbind(theta0, matrix(0, k, 2))
103   for(i in 1:k){
104     ## MH procedure for first parameter, using previous value for second
105     Y1 <- rgamma(1, theta[i,1], gbeta)
106     jj <- accfun1(Y1, theta[i,])
107     if(is.nan(jj)) jj <- -Inf  # adjust output for very small floats
108     TST1 <- log(runif(1)) <= jj
109     if(TST1) theta[i+1,1] <- Y1
110     else theta[i+1,1] <- theta[i,1]
111
112     ######################################################################
113     ## MH procedure for second parameter, using updated value for first
114
115     while(1){
116       Y2 <- rgamma(1, theta[i,2], gbeta)
117       if(Y2 <= 150)
118         break
119     }
120
121     jj <- accfun2(Y2, c(theta[i+1,1], theta[i,2]))
122     if(is.nan(jj)) jj <- -Inf  # adjust output for very small floats
123     TST2 <- log(runif(1)) <= jj
124     if(TST2) theta[i+1,2] <- Y2
125     else theta[i+1,2] <- theta[i,2]
126   }
127
128   return(theta)
129 }
130
131
132 bv_metrop_hast_gamma2 <- function(k, ffun, theta0 = c(1,1), gbeta = 1, ...)
133 {
134   ## see bv_metrop_hast_gamma()
135   ## when rejecting new realisations from proposal distribution
136   ## the previous accepted realisation is not repeated in the
137   ## current index (i.e. while instead of for)
138
139   accfun1 <- function(Y, X){
140     # function which determines if Y is accepted
141     # (for first parameter)
142     return( ffun(Y, X[2], ...) + dgamma(X[1], Y, gbeta, log=T) - ffun(X[1], X[2], ...) - dgamma(Y, X[1],
          gbeta, log=T) )
143   }
144   accfun2 <- function(Y, X){
145     # function which determines if Y is accepted
146     # (for second parameter)
147     return( ffun(X[1], Y, ...) + dgamma(X[2], Y, gbeta, log=T) - ffun(X[1], X[2], ...) - dgamma(Y, X[2],
          gbeta, log=T) )
148   }
149
150
```

```
  theta <- rbind(theta0, matrix(0, k, 2))
  for(i in 1:k){
    ## MH procedure for first parameter, using previous value for second
    while(1){
      Y1 <- rgamma(1, theta[i,1], gbeta)
      jj <- accfun1(Y1, theta[i,])
      if(is.nan(jj)) jj <- -Inf  # adjust output for very small floats
      TST1 <- log(runif(1)) <= jj
      if(TST1){
        theta[i+1,1] <- Y1
        break
      }
    }

    ######################################################################
    ## MH procedure for second parameter, using updated value for first

    while(1){
      while(1){
        Y2 <- rgamma(1, theta[i,2], gbeta)
        if(Y2 <= 150)
          break
      }

      jj <- accfun2(Y2, c(theta[i+1,1], theta[i,2]))
      if(is.nan(jj)) jj <- -Inf  # adjust output for very small floats
      TST2 <- log(runif(1)) <= jj
      if(TST2){
        theta[i+1,2] <- Y2
        break
      }
    }
  }

  return(theta)
}


bv_metrop_hast_gamma3 <- function(k, ffun, theta0=c(1,1), gbeta=1,
                                  min_vals=c(1e-5,1e-5), max_vals=c(150,150), ...)
{
  ## Function performing Metropolis-Hastings algorithm to generate
  ## Markov Chain with bivariate stationary distribution ffun.
  ## ffun must have variable as first argument and is specified in log form.
  ## Starting values is specified in theta0.
  ## Assumption of gamma proposal distribution, with first parameter
  ## to be updated and second (fixed) parameter specified in gbeta.
  ## '...' specifies optional arguments to ffun
  ## Output chain of length k+1 to account for initial values
  ## 'min_gen_val' is the minimum allowable candidate value for the second param

  accfun1 <- function(Y, X){
    # function which determines if Y is accepted
    # (for first parameter)
    return( ffun(Y, X[2], ...) + dgamma(X[1], Y, gbeta, log=T) - ffun(X[1], X[2], ...) - dgamma(Y, X[1],
     gbeta, log=T) )
  }
  accfun2 <- function(Y, X){
    # function which determines if Y is accepted
    # (for second parameter)
    return( ffun(X[1], Y, ...) + dgamma(X[2], Y, gbeta, log=T) - ffun(X[1], X[2], ...) - dgamma(Y, X[2],
     gbeta, log=T) )
  }


```

```
  theta <- rbind(theta0, matrix(0, k, 2))
  for(i in 1:k){
    ## MH procedure for first parameter, using previous value for second

    while(1){
      Y1 <- rgamma(1, theta[i,1], gbeta)
      if((min_vals[1] <= Y1) & (Y1 <= max_vals[1]))
        break
    }

    jj <- accfun1(Y1, theta[i,])
    if(is.nan(jj)) jj <- -Inf  # adjust output for very small floats
    TST1 <- log(runif(1)) <= jj
    if(TST1) theta[i+1,1] <- Y1
    else theta[i+1,1] <- theta[i,1]

    ######################################################################
    ## MH procedure for second parameter, using updated value for first

    while(1){
      Y2 <- rgamma(1, theta[i,2], gbeta)
      if((min_vals[2] <= Y2) & (Y2 <= max_vals[2]))
        break
    }

    jj <- accfun2(Y2, c(theta[i+1,1], theta[i,2]))
    if(is.nan(jj)) jj <- -Inf  # adjust output for very small floats
    TST2 <- log(runif(1)) <= jj
    if(TST2) theta[i+1,2] <- Y2
    else theta[i+1,2] <- theta[i,2]
  }

  return(theta)
}



######################################################################
########                                                      ########
########           MCMC METROPOLIS-HASTINGS SAMPLING          ########
########                FROM TRIVARIATE TARGET                ########
########                                                      ########
######################################################################


tv_metrop_hast_gamma <- function(k, ffun, theta0=c(1,1,1), gbeta=1,
                                 min_vals=c(1e-5,1e-5,1e-5), max_vals=c(150,150,150), ...)
{
  ## Function performing Metropolis-Hastings algorithm to generate
  ## Markov Chain with trivariate stationary distribution ffun.
  ## ffun must have variable as first argument and is specified in log form.
  ## Starting values is specified in theta0.
  ## Assumption of gamma proposal distribution, with first parameter
  ## to be updated and second (fixed) parameter specified in gbeta.
  ## '...' specifies optional arguments to ffun
  ## Output chain of length k+1 to account for initial values
  ## 'min_gen_val' is the minimum allowable candidate value for the second param


  accfun1 <- function(Y, X){
    # function which determines if Y is accepted
    # (for first parameter)
    return( ffun(Y, X[2], X[3], ...) + dgamma(X[1], Y, gbeta, log=T) - ffun(X[1], X[2], X[3], ...) - dgamma(Y
      , X[1], gbeta, log=T) )
  }
```

```
  accfun2 <- function(Y, X){
    # function which determines if Y is accepted
    # (for second parameter)
    return( ffun(X[1], Y, X[3], ...) + dgamma(X[2], Y, gbeta, log=T) - ffun(X[1], X[2], X[3], ...) - dgamma(Y
     , X[2], gbeta, log=T) )
  }
  accfun3 <- function(Y, X){
    # function which determines if Y is accepted
    # (for third parameter)
    return( ffun(X[1], X[2], Y, ...) + dgamma(X[3], Y, gbeta, log=T) - ffun(X[1], X[2], X[3], ...) - dgamma(Y
     , X[3], gbeta, log=T) )
  }

  theta <- rbind(theta0, matrix(0, k, 3))
  for(i in 1:k){
    cat(i, ":  ")
    ## MH procedure for first parameter, using previous value for second
    while(1){
      Y1 <- rgamma(1, theta[i,1], gbeta)
      if((min_vals[1] <= Y1) & (Y1 <= max_vals[1]))
        break
    }

    jj <- accfun1(Y1, theta[i,])
    if(is.nan(jj)) jj <- -Inf  # adjust output for very small floats
    TST1 <- log(runif(1)) <= jj
    if(TST1) theta[i+1,1] <- Y1
    else theta[i+1,1] <- theta[i,1]

    ######################################################################
    ## MH procedure for second parameter, using updated value for first

    while(1){
      Y2 <- rgamma(1, theta[i,2], gbeta)
      if((min_vals[2] <= Y2) & (Y2 <= max_vals[2]))
        break
    }

    jj <- accfun2(Y2, c(theta[i+1,1], theta[i,2:3]))
    if(is.nan(jj)) jj <- -Inf  # adjust output for very small floats
    TST2 <- log(runif(1)) <= jj
    if(TST2) theta[i+1,2] <- Y2
    else theta[i+1,2] <- theta[i,2]

    ######################################################################
    ## MH procedure for third parameter, using updated values

    while(1){
      Y3 <- rgamma(1, theta[i,3], gbeta)
      if((min_vals[3] <= Y3) & (Y3 <= max_vals[3]))
        break
    }

    jj <- accfun3(Y3, c(theta[i+1,1:2], theta[i,3]))
    if(is.nan(jj)) jj <- -Inf  # adjust output for very small floats
    TST3 <- log(runif(1)) <= jj
    if(TST3) theta[i+1,3] <- Y3
    else theta[i+1,3] <- theta[i,3]
  }

  return(theta)
}
```

B.3 R code for simulation studies

```
bayes_linex <- function(x, a)
{
  res <- NULL
  for(i in a){
    res <- c(res, -(1/i) * log( mean( exp(-i*x) ) ) )
  }
  return(res)
}


bayes_gent <- function(x, k)
{
  res <- NULL
  for(i in k){
    res <- c(res, (mean( x^(-i) ))^(-1/i) )
  }
  return(res)
}


mse_mae_bias <- function(est_vec, theta)
{
  n <- len(est_vec)
  mse <- mean((est_vec - theta)^2)
  mae <- mean(abs(est_vec - theta))
  bias <- mean(est_vec) - theta

  return(c(MAE=mae, MSE=mse, BIAS=bias))
}


bayes_est_fun <- function(y=cbind(0,nc=1), ts=seq(0,8,len=250), ffh, ffs, p_true,
                          a=c(-2,2,5), k=c(-2,2,5))
{
  n_a <- length(a)
  n_k <- length(k)
  emat_h <- matrix(0, 3+n_a+n_k, length(ts))
  emat_s <- emat_h

  p <- ncol(y)
  for(i in 1:length(ts)){
    if(p == 1){
      ftmp_h <- ffh(y, ts[i])
      ftmp_s <- ffs(y, ts[i])
    }
    else if(p == 2){
      ftmp_h <- ffh(y[,1], y[,2], ts[i])
      ftmp_s <- ffs(y[,1], y[,2], ts[i])
    }
    else if(p == 3){
      ftmp_h <- ffh(y[,1], y[,2], y[,3], ts[i])
      ftmp_s <- ffs(y[,1], y[,2], y[,3], ts[i])
    }

    emat_h[2,i] <- mean(ftmp_h)
    emat_h[3,i] <- median(ftmp_h)
    emat_s[2,i] <- mean(ftmp_s)
    emat_s[3,i] <- median(ftmp_s)

    for(j in 1:n_a){
      emat_h[3+j,i] <- -(1/a[j])*log( mean( exp(-a[j]*ftmp_h) ))
```

```
      emat_s[3+j,i] <- -(1/a[j])*log( mean( exp(-a[j]*ftmp_s) ))
    }


    for(j in 1:n_k){
      emat_h[3+n_a+j,i] <- (mean( ftmp_h^(-k[j]) ))^(-1/k[j])
      emat_s[3+n_a+j,i] <- (mean( ftmp_s^(-k[j]) ))^(-1/k[j])
    }
  }

  if(p == 1){
    ffh_true <- ffh(p_true, ts)
    ffs_true <- ffs(p_true, ts)
  }
  else if(p == 2){
    ffh_true <- ffh(p_true[1], p_true[2], ts)
    ffs_true <- ffs(p_true[1], p_true[2], ts)
  }
  else if(p == 3){
    ffh_true <- ffh(p_true[1], p_true[2], p_true[3], ts)
    ffs_true <- ffs(p_true[1], p_true[2], p_true[3], ts)
  }

  emat_h[1,] <- ffh_true
  emat_s[1,] <- ffs_true
  colnames(emat_h) <- colnames(emat_s) <- as.character(round(ts,3))
  rownames(emat_h) <- c("true_haz", "SE_loss", "AE_loss", paste0("LINEX_a=", a), paste0("GENT_k=", k))
  rownames(emat_s) <- c("true_surv", "SE_loss", "AE_loss", paste0("LINEX_a=", a), paste0("GENT_k=", k))

  return(list(emat_h, emat_s))
}


################################################################################
############  SIMULATION FUNCTIONS                    ##############################
################################################################################

simulator_crayexp <- function(n.samples, n, param, d, m=10000, burn=2500,
                              parvec, avec, kvec, print_i=FALSE, write_chains=FALSE,
                              read_chains=FALSE, read_path="chains/CRE/")
{
  tvec.fun <- gen_tvec_crayexp
  mhfun <- metrop_hast_gamma
  #ffn <- function(...) log(post_fun_crayexp2(...))
  ffn <- log_post_fun_crayexp

  # start timer
  time_start <- proc.time()[3]

  # matrix of samples with each column a generated sample of size 'n'
  if(!read_chains)
    sampmat <- replicate(n.samples, tvec.fun(n, param, d))
  else{
    fname <- paste0(read_path, "chains_",
                    paste("cre", paste0("g", param), n.samples, n, d, sep="_"), ".txt")
    cat("reading chain input file...\n")
    sampmat <- scan(fname, sep=",")
    sampmat <- t(matrix(sampmat, 1000, 10000, byrow=TRUE))
  }

  # matrix that will store MSE and bias corresponding to each
  # parameter value in 'parvec'
  parmat <- matrix(0, 2, length(parvec))
  rownames(parmat) <- c("MSE", "bias")
  colnames(parmat) <- as.character(parvec)
```

```
127
128  # matrix that will store Bayes linex estimator with varying 'a' parameter
129  ares <- numeric(length(avec))
130  names(ares) <- as.character(avec)
131
132  # matrix that will store Bayes GenEnt estimator with varying 'k' parameter
133  kres <- numeric(length(kvec))
134  names(kres) <- as.character(kvec)
135
136  # concatenate file name if specified
137  if(write_chains && (!read_chains))
138    fname <- paste0("~/Studies/RESEARCH/BAYESIAN SURVIVAL ANALYSIS/R stuff/BT/chains_",
139                    paste("cre", paste0("g", param), n.samples, n, d, sep="_"), ".txt")
140
141  # MH sampling to construct markov chain
142  cstats <- NULL
143  for(i in 1:ncol(sampmat)){
144    if(print_i)
145      cat(paste(round(100*i/n.samples, 1),"%\n"))
146
147    if(read_chains){
148      mc <- sampmat[,i]
149      if(length(mc) != m) stop("Incorrect number of values in column.")
150    }
151    else if(!read_chains){
152      x <- sampmat[,i]
153      mc <- mhfun(k=m+burn, ffun=ffn, tvec = x[-1], d = x[1])
154      mc <- mc[(burn+2):(burn+m+1)]
155
156      if(write_chains)
157        write.table(rbind(round(mc,7)), file=fname,
158                    append=T, quote=F, sep=",", row.names=F, col.names=F)
159    }
160
161
162    m_b <- mse_bias(mc, param)
163    cstats <- rbind(cstats, c(quantile(mc, probs = c(0.025, 0.975)),
164                              median = median(mc), mean = mean(mc), var = var(mc),
165                              mse=m_b[1], mae=mae(mc, param), bias=m_b[2]))
166
167    for(j in 1:length(parvec))
168      parmat[,j] <- parmat[,j] + mse_bias(mc, parvec[j])
169
170    ares <- ares + bayes_linex(mc, avec)
171    kres <- kres + bayes_gent(mc, kvec)
172  }
173
174  # calculate coverage for true parameter and constuct printing vector
175  coverage <- sum(apply(cstats[,1:2], 1,
176                        function(x) return((x[1] <= param) && (x[2] >= param)))) / n.samples
177  printvec <- round(c(cov=coverage, colMeans(cstats[,-(1:2)])), 4)
178
179
180  # process MSE, bias and coverage for parameters in 'parvec'
181  parmat <- rbind(parmat/n.samples, cov=rep(0, length(parvec)))
182  for(j in 1:length(parvec))
183    parmat[3,j] <- sum(apply(cstats[,1:2], 1,
184                             function(x) return((x[1] <= parvec[j]) && (x[2] >= parvec[j])))) / n.samples
185
186  # process Bayes linex and GenEnt estimator values
187  ares <- ares/n.samples
188  kres <- kres/n.samples
189
190  cat("\n\n")
191  cat(paste0(n.samples, " samples, size ", n, ", ", 1-d, "% censoring with parameters g=", param, "\n"))
```

```
  cat("GAMMA:\n") ; print(printvec)

  # end timer and print total runtime
  time_tot_s <- proc.time()[3] - time_start
  time_tot_h <- round(time_tot_s/60,1)
  cat("\ntotal time taken: ", time_tot_h, " minutes \n\n")

  rm(sampmat)

  return(list(chain_stats=cstats, print_stats=printvec,
              param_stats=parmat, a_result_vec=ares, k_result_vec=kres))
}


##############################################################################################
##############################################################################################


simulator_craygam <- function(id, n.samples, n, params, d, m=10000, burn=2500,
                              prior="jeff", parlist, alist, klist,
                              print_i=FALSE, write_chains=FALSE, read_chains=FALSE, read_path="chains/CRG/")
{
  tvec.fun <- gen_tvec_craygam
  mhfun <- bv_metrop_hast_gamma3
  #ffn <- function(...) log(post_fun_crayexp2(...))
  ffn <- log_post_fun_craygam

  # start timer
  time_start <- proc.time()[3]

  # matrix of samples with each column a generated sample of size 'n'
  if(!read_chains)
    sampmat <- replicate(n.samples, tvec.fun(n, params[1], params[2], d))
  else{
    fname_a <- paste0(read_path, "chains_",
                      paste("crg", id, prior, paste0("a", params[1]), n.samples, n, d, sep="_"), ".txt")
    fname_b <- paste0(read_path, "chains_",
                      paste("crg", id, prior, paste0("b", params[2]), n.samples, n, d, sep="_"), ".txt")

    cat("reading (alpha) chain input file...\n")
    sampmat_a <- scan(fname_a, sep=",")
    sampmat_a <- t(matrix(sampmat_a, 1000, 10000, byrow=TRUE))

    cat("reading (beta) chain input file...\n")
    sampmat_b <- scan(fname_b, sep=",")
    sampmat_b <- t(matrix(sampmat_b, 1000, 10000, byrow=TRUE))
  }


  parvec_a <- parlist[[1]]
  parvec_b <- parlist[[2]]
  avec_a <- alist[[1]]
  avec_b <- alist[[2]]
  kvec_a <- klist[[1]]
  kvec_b <- klist[[2]]

  # matrices that will store MSE and bias corresponding to each
  # parameter value in 'parlist'
  parmat_a <- matrix(0, 2, length(parvec_a))
  parmat_b <- matrix(0, 2, length(parvec_b))
  rownames(parmat_a) <- rownames(parmat_b) <- c("MSE", "bias")
  colnames(parmat_a) <- as.character(parvec_a)
  colnames(parmat_b) <- as.character(parvec_b)

  # matrix that will store Bayes linex estimator with varying 'a' parameter
```

```
  ares_a <- numeric(length(avec_a))
  names(ares_a) <- as.character(avec_a)
  ares_b <- numeric(length(avec_b))
  names(ares_b) <- as.character(avec_b)

  # matrix that will store Bayes GenEnt estimator with varying 'k' parameter
  kres_a <- numeric(length(kvec_a))
  names(kres_a) <- as.character(kvec_a)
  kres_b <- numeric(length(kvec_b))
  names(kres_b) <- as.character(kvec_b)

  # concatenate file name if specified
  if(write_chains && (!read_chains)){
    fname_a <- paste0("~/Studies/RESEARCH/BAYESIAN SURVIVAL ANALYSIS/R stuff/BT/chains_",
                      paste("crg", id, prior, paste0("a", params[1]), n.samples, n, d, sep="_"), ".txt")

    fname_b <- paste0("~/Studies/RESEARCH/BAYESIAN SURVIVAL ANALYSIS/R stuff/BT/chains_",
                      paste("crg", id, prior, paste0("b", params[2]), n.samples, n, d, sep="_"), ".txt")
  }

  # MH sampling to construct markov chain
  cstats_a <- NULL
  cstats_beta  <- NULL
  for(i in 1:n.samples){
    if(print_i)
      cat(paste(round(100*i/n.samples, 1),"%\n"))

    if(read_chains){
      mc1 <- sampmat_a[,i]
      if(length(mc1) != m) stop("Incorrect number of values in column (alpha).")

      mc2 <- sampmat_b[,i]
      if(length(mc2) != m) stop("Incorrect number of values in column (beta).")
    }
    else if(!read_chains){
      x <- sampmat[,i]
      mcmat <- mhfun(k=m+burn, ffun=ffn, tvec=x[-1], d=x[1], prior_t=prior)
      mc1 <- mcmat[(burn+2):(burn+m+1),1]
      mc2 <- mcmat[(burn+2):(burn+m+1),2]

      if(write_chains){
        write.table(rbind(round(mc1,7)), file=fname_a,
                    append=T, quote=F, sep=",", row.names=F, col.names=F)
        write.table(rbind(round(mc2,7)), file=fname_b,
                    append=T, quote=F, sep=",", row.names=F, col.names=F)
      }
    }

    # calculate results for ALPHA #
    mb_a <- mse_bias(mc1, params[1])
    cstats_a <- rbind(cstats_a, c(quantile(mc1, probs = c(0.025, 0.975)),
                                  median = median(mc1), mean = mean(mc1), var = var(mc1),
                                  mse=mb_a[1], mae=mae(mc1, params[1]), bias=mb_a[2]))

    for(j in 1:length(parvec_a))
      parmat_a[,j] <- parmat_a[,j] + mse_bias(mc1, parvec_a[j])

    ares_a <- ares_a + bayes_linex(mc1, avec_a)
    kres_a <- kres_a + bayes_gent(mc1, kvec_a)

    # calculate results for BETA #
    mb_b <- mse_bias(mc2, params[2])
    cstats_beta <- rbind(cstats_beta, c(quantile(mc2, probs = c(0.025, 0.975)),
                                        median = median(mc2), mean = mean(mc2), var = var(mc2),
                                        mse=mb_b[1], mae=mae(mc2, params[2]), bias=mb_b[2]))
```

```
322
323     for(j in 1:length(parvec_b))
324       parmat_b[,j] <- parmat_b[,j] + mse_bias(mc2, parvec_b[j])
325
326     ares_b <- ares_b + bayes_linex(mc2, avec_b)
327     kres_b <- kres_b + bayes_gent(mc2, kvec_b)
328
329   }
330
331   # pool results over all samples for ALPHA #
332   cov_alpha <- sum(apply(cstats_a[,1:2], 1,
333                          function(x) return((x[1] <= params[1]) && (x[2] >= params[1])))) / n.samples
334   printvec_a <- round(c(cov=cov_alpha, colMeans(cstats_a[,-(1:2)])), 4)
335
336   ares_a <- ares_a/n.samples
337   kres_a <- kres_a/n.samples
338   parmat_a <- rbind(parmat_a/n.samples, cov=rep(0, length(parvec_a)))
339   for(j in 1:length(parvec_a))
340     parmat_a[3,j] <- sum(apply(cstats_a[,1:2], 1,
341                                function(x) return((x[1] <= parvec_a[j]) && (x[2] >= parvec_a[j])))) / n.
       samples
342
343   # pool results over all samples for BETA #
344   cov_beta <- sum(apply(cstats_beta[,1:2], 1,
345                         function(x) return((x[1] <= params[2]) && (x[2] >= params[2])))) / n.samples
346   printvec_b <- round(c(cov=cov_beta, colMeans(cstats_beta[,-(1:2)])), 4)
347
348   ares_b <- ares_b/n.samples
349   kres_b <- kres_b/n.samples
350   parmat_b <- rbind(parmat_b/n.samples, cov=rep(0, length(parvec_b)))
351   for(j in 1:length(parvec_b))
352     parmat_b[3,j] <- sum(apply(cstats_beta[,1:2], 1,
353                                function(x) return((x[1] <= parvec_b[j]) && (x[2] >= parvec_b[j])))) / n.
       samples
354
355
356   cat(paste0(n.samples, " samples, size ", n, ", ", 1-d, "% censoring\nwith parameters a=", params[1], " and
       b=", params[2],"\n"))
357   cat("ALPHA:\n") ; print(printvec_a) ; cat("\nBETA:\n") ; print(printvec_b)
358
359   # end timer and print total runtime
360   time_tot_s <- proc.time()[3] - time_start
361   time_tot_h <- round(time_tot_s/60,1)
362   cat("\ntotal time taken: ", time_tot_h, " minutes \n\n")
363
364   rm(sampmat_a)
365   rm(sampmat_b)
366
367   return(list(cs_alpha=cstats_a, cs_beta=cstats_beta,
368               prstats_alpha=printvec_a, prstats_beta=printvec_b,
369               parstats_alpha=parmat_a, parstats_beta=parmat_b,
370               a_results_alpha=ares_a, a_results_beta=ares_b,
371               k_results_alpha=kres_a, k_results_beta=kres_b))
372 }
373
374
375 ##############################################################################################
376 ##############################################################################################
377
378
379 simulator_gcrayexp <- function(id, n.samples, n, params, d, m=10000, burn=2500,
380                                parlist, alist, klist, print_i=FALSE,
381                                write_chains=FALSE, read_chains=FALSE, read_path="chains/GCRE/")
382 {
383   tvec.fun <- gen_tvec_gcrayexp
```

```
mhfun <- bv_metrop_hast_gamma3
#ffn <- function(...) log(post_fun_crayexp2(...))
ffn <- log_post_fun_gcrayexp

# start timer
time_start <- proc.time()[3]

# matrix of samples with each column a generated sample of size 'n'
if(!read_chains)
  sampmat <- replicate(n.samples, tvec.fun(n, params[1], params[2], d))
else{
  fname_g <- paste0(read_path, "chains_",
                    paste("gcre", id, paste0("g", params[1]), n.samples, n, d, sep="_"), ".txt")
  fname_c <- paste0(read_path, "chains_",
                    paste("gcre", id, paste0("c", params[2]), n.samples, n, d, sep="_"), ".txt")

  cat("reading (gamma) chain input file...\n")
  sampmat_g <- scan(fname_g, sep=",")
  sampmat_g <- t(matrix(sampmat_g, 1000, 10000, byrow=TRUE))

  cat("reading (c) chain input file...\n")
  sampmat_c <- scan(fname_c, sep=",")
  sampmat_c <- t(matrix(sampmat_c, 1000, 10000, byrow=TRUE))
}

parvec_g <- parlist[[1]]
parvec_c <- parlist[[2]]
avec_g <- alist[[1]]
avec_c <- alist[[2]]
kvec_g <- klist[[1]]
kvec_c <- klist[[2]]

# matrices that will store MSE and bias corresponding to each
# parameter value in 'parlist'
parmat_g <- matrix(0, 2, length(parvec_g))
parmat_c <- matrix(0, 2, length(parvec_c))
rownames(parmat_g) <- rownames(parmat_c) <- c("MSE", "bias")
colnames(parmat_g) <- as.character(parvec_g)
colnames(parmat_c) <- as.character(parvec_c)

# matrix that will store Bayes linex estimator with varying 'a' parameter
ares_g <- numeric(length(avec_g))
names(ares_g) <- as.character(avec_g)
ares_c <- numeric(length(avec_c))
names(ares_c) <- as.character(avec_c)

# matrix that will store Bayes GenEnt estimator with varying 'k' parameter
kres_g <- numeric(length(kvec_g))
names(kres_g) <- as.character(kvec_g)
kres_c <- numeric(length(kvec_c))
names(kres_c) <- as.character(kvec_c)

# concatenate file name if specified
if(write_chains && (!read_chains)){
  fname_g <- paste0("~/Studies/RESEARCH/BAYESIAN SURVIVAL ANALYSIS/R stuff/BT/chains_",
                    paste("gcre", id, paste0("g", params[1]), n.samples, n, d, sep="_"), ".txt")

  fname_c <- paste0("~/Studies/RESEARCH/BAYESIAN SURVIVAL ANALYSIS/R stuff/BT/chains_",
                    paste("gcre", id, paste0("c", params[2]), n.samples, n, d, sep="_"), ".txt")
}

# MH sampling to construct markov chain
cstats_gam <- NULL
cstats_cc  <- NULL
for(i in 1:n.samples){
```

```
449     if(print_i)
450       cat(paste(round(100*i/n.samples, 1),"%\n"))
451 
452     if(read_chains){
453       mc1 <- sampmat_g[,i]
454       if(length(mc1) != m) stop("Incorrect number of values in column (gamma).")
455 
456       mc2 <- sampmat_c[,i]
457       if(length(mc2) != m) stop("Incorrect number of values in column (c).")
458     }
459     else if(!read_chains){
460       x <- sampmat[,i]
461       mcmat <- mhfun(k=m+burn, ffun=ffn, theta0=c(1,2), min_vals=c(1e-5,1), tvec=x[-1], d=x[1])
462       mc1 <- mcmat[(burn+2):(burn+m+1),1]
463       mc2 <- mcmat[(burn+2):(burn+m+1),2]
464 
465       if(write_chains){
466         write.table(rbind(round(mc1,7)), file=fname_g,
467                     append=T, quote=F, sep=",", row.names=F, col.names=F)
468         write.table(rbind(round(mc2,7)), file=fname_c,
469                     append=T, quote=F, sep=",", row.names=F, col.names=F)
470       }
471     }
472 
473     # calculate results for ALPHA #
474     mb_g <- mse_bias(mc1, params[1])
475     cstats_gam <- rbind(cstats_gam, c(quantile(mc1, probs = c(0.025, 0.975)),
476                                       median = median(mc1), mean = mean(mc1), var = var(mc1),
477                                       mse=mb_g[1], mae=mae(mc1, params[1]), bias=mb_g[2]))
478 
479     for(j in 1:length(parvec_g))
480       parmat_g[,j] <- parmat_g[,j] + mse_bias(mc1, parvec_g[j])
481 
482     ares_g <- ares_g + bayes_linex(mc1, avec_g)
483     kres_g <- kres_g + bayes_gent(mc1, kvec_g)
484 
485 
486     # calculate results for BETA #
487     mb_c <- mse_bias(mc2, params[2])
488     cstats_cc <- rbind(cstats_cc, c(quantile(mc2, probs = c(0.025, 0.975)),
489                                     median = median(mc2), mean = mean(mc2), var = var(mc2),
490                                     mse=mb_c[1], mae=mae(mc2, params[2]), bias=mb_c[2]))
491 
492     for(j in 1:length(parvec_c))
493       parmat_c[,j] <- parmat_c[,j] + mse_bias(mc2, parvec_c[j])
494 
495     ares_c <- ares_c + bayes_linex(mc2, avec_c)
496     kres_c <- kres_c + bayes_gent(mc2, kvec_c)
497 
498   }
499 
500   # pool results over all samples for ALPHA #
501   cov_gam <- sum(apply(cstats_gam[,1:2], 1,
502                        function(x) return((x[1] <= params[1]) && (x[2] >= params[1])))) / n.samples
503   printvec_g <- round(c(cov=cov_gam, colMeans(cstats_gam[,-(1:2)])), 4)
504 
505   ares_g <- ares_g/n.samples
506   kres_g <- kres_g/n.samples
507   parmat_g <- rbind(parmat_g/n.samples, cov=rep(0, length(parvec_g)))
508   for(j in 1:length(parvec_g))
509     parmat_g[3,j] <- sum(apply(cstats_gam[,1:2], 1,
510                                function(x) return((x[1] <= parvec_g[j]) && (x[2] >= parvec_g[j])))) / n.
      samples
511 
512   # pool results over all samples for BETA #
```

```
  cov_cc <- sum(apply(cstats_cc[,1:2], 1,
                      function(x) return((x[1] <= params[2]) && (x[2] >= params[2])))) / n.samples
  printvec_c <- round(c(cov=cov_cc, colMeans(cstats_cc[,-(1:2)])), 4)

  ares_c <- ares_c/n.samples
  kres_c <- kres_c/n.samples
  parmat_c <- rbind(parmat_c/n.samples, cov=rep(0, length(parvec_c)))
  for(j in 1:length(parvec_c))
    parmat_c[3,j] <- sum(apply(cstats_cc[,1:2], 1,
                               function(x) return((x[1] <= parvec_c[j]) && (x[2] >= parvec_c[j])))) / n.
     samples

  cat(paste0(n.samples, " samples, size ", n, ", ", 1-d, "% censoring\nwith parameters g=", params[1], " and
     c=", params[2],"\n"))
  cat("GAMMA:\n") ; print(printvec_g) ; cat("\nC:\n") ; print(printvec_c)

  # end timer and print total runtime
  time_tot_s <- proc.time()[3] - time_start
  time_tot_h <- round(time_tot_s/60,1)
  cat("\ntotal time taken: ", time_tot_h, " minutes \n\n")

  rm(sampmat_g)
  rm(sampmat_c)

  return(list(cs_gamma=cstats_gam, cs_c=cstats_cc,
              prstats_gamma=printvec_g, prstats_c=printvec_c,
              parstats_gamma=parmat_g, parstats_c=parmat_c,
              a_results_gamma=ares_g, a_results_c=ares_c,
              k_results_gamma=kres_g, k_results_c=kres_c))
}



################################################################################################
################################################################################################


simulator_gcraygam <- function(id, n.samples, n, params, d, m=10000, burn=2500, prior="jeff",
                               parlist, alist, klist, print_i=FALSE, write_chains=FALSE,
                               read_chains=FALSE, write_samples=TRUE, read_path="chains/GCRG/",
                               mh_maxvals=c(150,150,150))
{
  tvec.fun <- gen_tvec_gcraygam
  mhfun <- tv_metrop_hast_gamma
  ffn <- log_post_fun_gcraygam_cond

  # start timer
  time_start <- proc.time()[3]

  # matrix of samples with each column a generated sample of size 'n'
  if(!read_chains){
    sampmat <- replicate(n.samples, tvec.fun(n, params[1], params[2], params[3], d))
    if(write_samples){
      fname_s <- paste0("samples_",
                        paste("gcrg", id, prior, paste0("a", params[1]), n.samples, n, d, sep="_"), ".txt")
      write.table( round(t(sampmat),7), file=fname_s, sep=",", quote=F, row.names=F, col.names=F)
    }
  }
  else{
    fname_a <- paste0(read_path, "chains_",
                      paste("gcrg", id, prior, paste0("a", params[1]), n.samples, n, d, sep="_"), ".txt")
    fname_b <- paste0(read_path, "chains_",
                      paste("gcrg", id, prior, paste0("b", params[2]), n.samples, n, d, sep="_"), ".txt")
    fname_c <- paste0(read_path, "chains_",
                      paste("gcrg", id, prior, paste0("c", params[3]), n.samples, n, d, sep="_"), ".txt")
```

```
576
577     cat("reading (alpha) chain input file...\n")
578     sampmat_a <- scan(fname_a, sep=",")
579     if(len(sampmat_a) > 10000000) sampmat_a <- sampmat_a[1:10000000]
580     sampmat_a <- t(matrix(sampmat_a, 1000, 10000, byrow=TRUE))
581
582     cat("reading (beta) chain input file...\n")
583     sampmat_b <- scan(fname_b, sep=",")
584     if(len(sampmat_b) > 10000000) sampmat_b <- sampmat_b[1:10000000]
585     sampmat_b <- t(matrix(sampmat_b, 1000, 10000, byrow=TRUE))
586
587     cat("reading (c) chain input file...\n")
588     sampmat_c <- scan(fname_c, sep=",")
589     if(len(sampmat_c) > 10000000) sampmat_c <- sampmat_c[1:10000000]
590     sampmat_c <- t(matrix(sampmat_c, 1000, 10000, byrow=TRUE))
591   }
592
593   parvec_a <- parlist[[1]]
594   parvec_b <- parlist[[2]]
595   parvec_c <- parlist[[3]]
596
597   avec_a <- alist[[1]]
598   avec_b <- alist[[2]]
599   avec_c <- alist[[3]]
600
601   kvec_a <- klist[[1]]
602   kvec_b <- klist[[2]]
603   kvec_c <- klist[[3]]
604
605   # matrices that will store MSE and bias corresponding to each
606   # parameter value in 'parlist'
607   parmat_a <- matrix(0, 2, length(parvec_a))
608   parmat_b <- matrix(0, 2, length(parvec_b))
609   parmat_c <- matrix(0, 2, length(parvec_c))
610   rownames(parmat_a) <- rownames(parmat_b) <- rownames(parmat_c) <- c("MSE", "bias")
611   colnames(parmat_a) <- as.character(parvec_a)
612   colnames(parmat_b) <- as.character(parvec_b)
613   colnames(parmat_c) <- as.character(parvec_c)
614
615   # matrix that will store Bayes linex estimator with varying 'a' parameter
616   ares_a <- numeric(length(avec_a))
617   names(ares_a) <- as.character(avec_a)
618   ares_b <- numeric(length(avec_b))
619   names(ares_b) <- as.character(avec_b)
620   ares_c <- numeric(length(avec_c))
621   names(ares_c) <- as.character(avec_c)
622
623   # matrix that will store Bayes GenEnt estimator with varying 'k' parameter
624   kres_a <- numeric(length(kvec_a))
625   names(kres_a) <- as.character(kvec_a)
626   kres_b <- numeric(length(kvec_b))
627   names(kres_b) <- as.character(kvec_b)
628   kres_c <- numeric(length(kvec_c))
629   names(kres_c) <- as.character(kvec_c)
630
631   # concatenate file name if specified
632   if(write_chains && (!read_chains)){
633     fname_a <- paste0("~/Studies/RESEARCH/BAYESIAN SURVIVAL ANALYSIS/R stuff/BT/chains_",
634                       paste("gcrg", id, prior, paste0("a", params[1]), n.samples, n, d, sep="_"), ".txt")
635
636     fname_b <- paste0("~/Studies/RESEARCH/BAYESIAN SURVIVAL ANALYSIS/R stuff/BT/chains_",
637                       paste("gcrg", id, prior, paste0("b", params[2]), n.samples, n, d, sep="_"), ".txt")
638
639     fname_c <- paste0("~/Studies/RESEARCH/BAYESIAN SURVIVAL ANALYSIS/R stuff/BT/chains_",
640                       paste("gcrg", id, prior, paste0("c", params[3]), n.samples, n, d, sep="_"), ".txt")
```

```
}

# MH sampling to construct markov chain
cstats_a  <- NULL
cstats_b  <- NULL
cstats_cc <- NULL
for(i in 1:n.samples){
  if(print_i)
    cat(paste(round(100*i/n.samples, 1),"%\n")

  if(read_chains){
    mc1 <- sampmat_a[,i]
    if(length(mc1) != m) stop("Incorrect number of values in column (alpha).")

    mc2 <- sampmat_b[,i]
    if(length(mc2) != m) stop("Incorrect number of values in column (beta).")

    mc3 <- sampmat_c[,i]
    if(length(mc3) != m) stop("Incorrect number of values in column (c).")
  }
  else if(!read_chains){
    x <- sampmat[,i]
    mcmat <- mhfun(k=m+burn, ffun=ffn, theta0=c(1,1,2), min_vals=c(1e-5,1e-05,1), max_vals=mh_maxvals, tvec
   =x[-1], d=x[1], prior_t=prior)
    mc1 <- mcmat[(burn+2):(burn+m+1),1]
    mc2 <- mcmat[(burn+2):(burn+m+1),2]
    mc3 <- mcmat[(burn+2):(burn+m+1),3]

    if(write_chains){
      write.table(rbind(round(mc1,7)), file=fname_a,
                  append=T, quote=F, sep=",", row.names=F, col.names=F)
      write.table(rbind(round(mc2,7)), file=fname_b,
                  append=T, quote=F, sep=",", row.names=F, col.names=F)
      write.table(rbind(round(mc3,7)), file=fname_c,
                  append=T, quote=F, sep=",", row.names=F, col.names=F)
    }
  }

  # calculate results for ALPHA #
  mb_a <- mse_bias(mc1, params[1])
  cstats_a <- rbind(cstats_a, c(quantile(mc1, probs = c(0.025, 0.975)),
                                median = median(mc1), mean = mean(mc1), var = var(mc1),
                                mse=mb_a[1], mae=mae(mc1, params[1]), bias=mb_a[2]))

  for(j in 1:length(parvec_a))
    parmat_a[,j] <- parmat_a[,j] + mse_bias(mc1, parvec_a[j])

  ares_a <- ares_a + bayes_linex(mc1, avec_a)
  kres_a <- kres_a + bayes_gent(mc1, kvec_a)

  # calculate results for BETA #
  mb_b <- mse_bias(mc2, params[2])
  cstats_b <- rbind(cstats_b, c(quantile(mc2, probs = c(0.025, 0.975)),
                                median = median(mc2), mean = mean(mc2), var = var(mc2),
                                mse=mb_b[1], mae=mae(mc2, params[2]), bias=mb_b[2]))

  for(j in 1:length(parvec_b))
    parmat_b[,j] <- parmat_b[,j] + mse_bias(mc2, parvec_b[j])

  ares_b <- ares_b + bayes_linex(mc2, avec_b)
  kres_b <- kres_b + bayes_gent(mc2, kvec_b)

  # calculate results for CC #
  mb_c <- mse_bias(mc3, params[3])
  cstats_cc <- rbind(cstats_cc, c(quantile(mc3, probs = c(0.025, 0.975)),
```

```
                                    median = median(mc3), mean = mean(mc3), var = var(mc3),
                                    mse=mb_c[1], mae=mae(mc3, params[3]), bias=mb_c[2]))

  for(j in 1:length(parvec_c))
    parmat_c[,j] <- parmat_c[,j] + mse_bias(mc3, parvec_c[j])

  ares_c <- ares_c + bayes_linex(mc3, avec_c)
  kres_c <- kres_c + bayes_gent(mc3, kvec_c)

}

# pool results over all samples for ALPHA #
cov_alpha <- sum(apply(cstats_a[,1:2], 1,
                       function(x) return((x[1] <= params[1]) && (x[2] >= params[1])))) / n.samples
printvec_a <- round(c(cov=cov_alpha, colMeans(cstats_a[,-(1:2)])), 4)

ares_a <- ares_a/n.samples
kres_a <- kres_a/n.samples
parmat_a <- rbind(parmat_a/n.samples, cov=rep(0, length(parvec_a)))
for(j in 1:length(parvec_a))
  parmat_a[3,j] <- sum(apply(cstats_a[,1:2], 1,
                             function(x) return((x[1] <= parvec_a[j]) && (x[2] >= parvec_a[j])))) / n.
    samples

# pool results over all samples for BETA #
cov_beta <- sum(apply(cstats_b[,1:2], 1,
                      function(x) return((x[1] <= params[2]) && (x[2] >= params[2])))) / n.samples
printvec_b <- round(c(cov=cov_beta, colMeans(cstats_b[,-(1:2)])), 4)

ares_b <- ares_b/n.samples
kres_b <- kres_b/n.samples
parmat_b <- rbind(parmat_b/n.samples, cov=rep(0, length(parvec_b)))
for(j in 1:length(parvec_b))
  parmat_b[3,j] <- sum(apply(cstats_b[,1:2], 1,
                             function(x) return((x[1] <= parvec_b[j]) && (x[2] >= parvec_b[j])))) / n.
    samples

# pool results over all samples for CC #
cov_c <- sum(apply(cstats_cc[,1:2], 1,
                   function(x) return((x[1] <= params[3]) && (x[2] >= params[3])))) / n.samples
printvec_c <- round(c(cov=cov_c, colMeans(cstats_cc[,-(1:2)])), 4)

ares_c <- ares_c/n.samples
kres_c <- kres_c/n.samples
parmat_c <- rbind(parmat_c/n.samples, cov=rep(0, length(parvec_c)))
for(j in 1:length(parvec_c))
  parmat_c[3,j] <- sum(apply(cstats_cc[,1:2], 1,
                             function(x) return((x[1] <= parvec_c[j]) && (x[2] >= parvec_c[j])))) / n.
    samples

cat(paste0(n.samples, " samples, size ", n, ", ", 1-d, "% censoring\nwith parameters a=", params[1], ", b="
    , params[2], " and c=", params[3], "\n"))
cat("ALPHA:\n") ; print(printvec_a) ; cat("\nBETA:\n") ; print(printvec_b) ; cat("\nC:\n") ; print(printvec
    _c)

# end timer and print total runtime
time_tot_s <- proc.time()[3] - time_start
time_tot_h <- round(time_tot_s/60,1)
cat("\ntotal time taken: ", time_tot_h, " minutes \n\n")

rm(sampmat_a)
rm(sampmat_b)
rm(sampmat_c)
```

```
  return(list(cs_alpha=cstats_a,        cs_beta=cstats_b,     cs_c=cstats_cc,
              prstats_alpha=printvec_a, prstats_beta=printvec_b, prstats_c=printvec_c,
              parstats_alpha=parmat_a,  parstats_beta=parmat_b,  parstats_c=parmat_c,
              a_results_alpha=ares_a,   a_results_beta=ares_b,   a_results_c=ares_c,
              k_results_alpha=kres_a,   k_results_beta=kres_b,   k_results_c=kres_c))
}


##############################################################################################
##############################################################################################


simulator_crayexp_asym <- function(n.samples, n, param, d, m=10000, burn=2500,
                                   avec, kvec, read_path="chains/CRE/")
{
  fname <- paste0(read_path, "chains_",
                  paste("cre", paste0("g", param), n.samples, n, d, sep="_"), ".txt")
  cat("reading chain input file...\n")
  sampmat <- scan(fname, sep=",")
  sampmat <- t(matrix(sampmat, 1000, 10000, byrow=TRUE))

  # matrix that will store Bayes linex estimator with varying 'a' parameter
  ares <- numeric(length(avec))
  names(ares) <- as.character(avec)

  # matrix that will store Bayes GenEnt estimator with varying 'k' parameter
  kres <- numeric(length(kvec))
  names(kres) <- as.character(kvec)

  for(i in 1:ncol(sampmat)){
    mc <- sampmat[,i]
    if(length(mc) != m) stop("Incorrect number of values in column.")

    ares <- rbind(ares, bayes_linex(mc, avec))
    kres <- rbind(kres, bayes_gent(mc, kvec))
  }

  rm(sampmat)

  return(list(a_result_vec=ares, k_result_vec=kres))
}


##############################################################################################


simulator_craygam_asym <- function(id, n.samples, n, params, d, m=10000, burn=2500,
                                   prior="jeff", alist, klist, read_path="chains/CRG/")
{
  fname_a <- paste0(read_path, "chains_",
                    paste("crg", id, prior, paste0("a", params[1]), n.samples, n, d, sep="_"), ".txt")
  fname_b <- paste0(read_path, "chains_",
                    paste("crg", id, prior, paste0("b", params[2]), n.samples, n, d, sep="_"), ".txt")

  cat("reading (alpha) chain input file...\n")
  sampmat_a <- scan(fname_a, sep=",")
  sampmat_a <- t(matrix(sampmat_a, 1000, 10000, byrow=TRUE))

  cat("reading (beta) chain input file...\n")
  sampmat_b <- scan(fname_b, sep=",")
  sampmat_b <- t(matrix(sampmat_b, 1000, 10000, byrow=TRUE))

  avec_a <- alist[[1]]
  avec_b <- alist[[2]]
  kvec_a <- klist[[1]]
```

```
  kvec_b <- klist[[2]]

  # matrix that will store Bayes linex estimator with varying 'a' parameter
  ares_a <- numeric(length(avec_a))
  names(ares_a) <- as.character(avec_a)
  ares_b <- numeric(length(avec_b))
  names(ares_b) <- as.character(avec_b)

  # matrix that will store Bayes GenEnt estimator with varying 'k' parameter
  kres_a <- numeric(length(kvec_a))
  names(kres_a) <- as.character(kvec_a)
  kres_b <- numeric(length(kvec_b))
  names(kres_b) <- as.character(kvec_b)

  # MH sampling to construct markov chain
  for(i in 1:n.samples){
    mc1 <- sampmat_a[,i]
    if(length(mc1) != m) stop("Incorrect number of values in column (alpha).")

    mc2 <- sampmat_b[,i]
    if(length(mc2) != m) stop("Incorrect number of values in column (beta).")

    ares_a <- rbind(ares_a, bayes_linex(mc1, avec_a))
    kres_a <- rbind(kres_a, bayes_gent(mc1, kvec_a))

    ares_b <- rbind(ares_b, bayes_linex(mc2, avec_b))
    kres_b <- rbind(kres_b, bayes_gent(mc2, kvec_b))

  }

  rm(sampmat_a)
  rm(sampmat_b)

  return(list(a_results_alpha=ares_a, a_results_beta=ares_b,
              k_results_alpha=kres_a, k_results_beta=kres_b))
}


##############################################################################################

simulator_gcrayexp_asym <- function(id, n.samples, n, params, d, m=10000, burn=2500,
                                    alist, klist, read_path="chains/GCRE/")
{
  fname_g <- paste0(read_path, "chains_",
                    paste("gcre", id, paste0("g", params[1]), n.samples, n, d, sep="_"), ".txt")
  fname_c <- paste0(read_path, "chains_",
                    paste("gcre", id, paste0("c", params[2]), n.samples, n, d, sep="_"), ".txt")

  cat("reading (gamma) chain input file...\n")
  sampmat_g <- scan(fname_g, sep=",")
  sampmat_g <- t(matrix(sampmat_g, 1000, 10000, byrow=TRUE))

  cat("reading (c) chain input file...\n")
  sampmat_c <- scan(fname_c, sep=",")
  sampmat_c <- t(matrix(sampmat_c, 1000, 10000, byrow=TRUE))

  avec_g <- alist[[1]]
  avec_c <- alist[[2]]
  kvec_g <- klist[[1]]
  kvec_c <- klist[[2]]

  # matrix that will store Bayes linex estimator with varying 'a' parameter
  ares_g <- numeric(length(avec_g))
  names(ares_g) <- as.character(avec_g)
```

```
  ares_c <- numeric(length(avec_c))
  names(ares_c) <- as.character(avec_c)

  # matrix that will store Bayes GenEnt estimator with varying 'k' parameter
  kres_g <- numeric(length(kvec_g))
  names(kres_g) <- as.character(kvec_g)
  kres_c <- numeric(length(kvec_c))
  names(kres_c) <- as.character(kvec_c)

  # MH sampling to construct markov chain
  for(i in 1:n.samples){
    mc1 <- sampmat_g[,i]
    if(length(mc1) != m) stop("Incorrect number of values in column (gamma).")

    mc2 <- sampmat_c[,i]
    if(length(mc2) != m) stop("Incorrect number of values in column (c).")

    ares_g <- rbind(ares_g, bayes_linex(mc1, avec_g))
    kres_g <- rbind(kres_g, bayes_gent(mc1, kvec_g))

    ares_c <- rbind(ares_c, bayes_linex(mc2, avec_c))
    kres_c <- rbind(kres_c, bayes_gent(mc2, kvec_c))

  }

  rm(sampmat_g)
  rm(sampmat_c)

  return(list(a_results_gamma=ares_g, a_results_c=ares_c,
              k_results_gamma=kres_g, k_results_c=kres_c))
}


##############################################################################################


simulator_gcraygam_asym <- function(id, n.samples, n, params, d, m=10000, burn=2500, prior="jeff",
                                alist, klist, read_path="chains/GCRG/")
{
  fname_a <- paste0(read_path, "chains_",
                    paste("gcrg", id, prior, paste0("a", params[1]), n.samples, n, d, sep="_"), ".txt")
  fname_b <- paste0(read_path, "chains_",
                    paste("gcrg", id, prior, paste0("b", params[2]), n.samples, n, d, sep="_"), ".txt")
  fname_c <- paste0(read_path, "chains_",
                    paste("gcrg", id, prior, paste0("c", params[3]), n.samples, n, d, sep="_"), ".txt")

  cat("reading (alpha) chain input file...\n")
  sampmat_a <- scan(fname_a, sep=",")
  if(len(sampmat_a) > 10000000) sampmat_a <- sampmat_a[1:10000000]
  sampmat_a <- t(matrix(sampmat_a, 1000, 10000, byrow=TRUE))

  cat("reading (beta) chain input file...\n")
  sampmat_b <- scan(fname_b, sep=",")
  if(len(sampmat_b) > 10000000) sampmat_b <- sampmat_b[1:10000000]
  sampmat_b <- t(matrix(sampmat_b, 1000, 10000, byrow=TRUE))

  cat("reading (c) chain input file...\n")
  sampmat_c <- scan(fname_c, sep=",")
  if(len(sampmat_c) > 10000000) sampmat_c <- sampmat_c[1:10000000]
  sampmat_c <- t(matrix(sampmat_c, 1000, 10000, byrow=TRUE))

  avec_a <- alist[[1]]
  avec_b <- alist[[2]]
  avec_c <- alist[[3]]

```

```
  kvec_a <- klist[[1]]
  kvec_b <- klist[[2]]
  kvec_c <- klist[[3]]

  # matrix that will store Bayes linex estimator with varying 'a' parameter
  ares_a <- numeric(length(avec_a))
  names(ares_a) <- as.character(avec_a)
  ares_b <- numeric(length(avec_b))
  names(ares_b) <- as.character(avec_b)
  ares_c <- numeric(length(avec_c))
  names(ares_c) <- as.character(avec_c)

  # matrix that will store Bayes GenEnt estimator with varying 'k' parameter
  kres_a <- numeric(length(kvec_a))
  names(kres_a) <- as.character(kvec_a)
  kres_b <- numeric(length(kvec_b))
  names(kres_b) <- as.character(kvec_b)
  kres_c <- numeric(length(kvec_c))
  names(kres_c) <- as.character(kvec_c)

  # MH sampling to construct markov chain
  for(i in 1:n.samples){
    mc1 <- sampmat_a[,i]
    if(length(mc1) != m) stop("Incorrect number of values in column (alpha).")

    mc2 <- sampmat_b[,i]
    if(length(mc2) != m) stop("Incorrect number of values in column (beta).")

    mc3 <- sampmat_c[,i]
    if(length(mc3) != m) stop("Incorrect number of values in column (c).")

    ares_a <- c(ares_a, bayes_linex(mc1, avec_a))
    kres_a <- c(kres_a, bayes_gent(mc1, kvec_a))

    ares_b <- c(ares_b, bayes_linex(mc2, avec_b))
    kres_b <- c(kres_b, bayes_gent(mc2, kvec_b))

    ares_c <- c(ares_c, bayes_linex(mc3, avec_c))
    kres_c <- c(kres_c, bayes_gent(mc3, kvec_c))

  }

  rm(sampmat_a)
  rm(sampmat_b)
  rm(sampmat_c)

  return(list(a_results_alpha=ares_a,   a_results_beta=ares_b,   a_results_c=ares_c,
              k_results_alpha=kres_a,   k_results_beta=kres_b,   k_results_c=kres_c))
}
```

B.4 R code for plots

```
library(ggplot2)

bayes_est_ggplot_cens <- function(sim1, sim2, model, param_true, fname, x_nbreaks=10, limx, limy,
                                  show_plot=TRUE, win_plot=TRUE, width=14, height=6)
{
  if(model == "cre"){
    labx   <- expression(paste(gamma, " parameter value"))
    a_ind <- 4 ; p_ind <- 2 ; k_ind <- 5
  }
```

```
  else if(model == "gcreG"){
    labx   <- expression(paste(gamma, " parameter value"))
    a_ind <- 7 ; p_ind <- 3 ; k_ind <- 9
  }
  else if(model == "gcreC"){
    labx   <- expression(paste(italic(c), " parameter value"))
    a_ind <- 8 ; p_ind <- 4 ; k_ind <- 10
  }
  else
    stop("Details incorrectly specified.")

  aresv1 <- sim1[[a_ind]]
  aresv2 <- sim2[[a_ind]]
  kresv1 <- sim1[[k_ind]]
  kresv2 <- sim2[[k_ind]]
  pstats1 <- sim1[[p_ind]]
  pstats2 <- sim2[[p_ind]]

  means <- c(pstats1[3], pstats2[3])
  meds <- c(pstats1[2], pstats2[2])

  len <- function(...) length(...)
  pdat <- data.frame(y=c(as.numeric(names(aresv1)), as.numeric(names(aresv2)),
                         as.numeric(names(kresv1)), as.numeric(names(kresv2))),
                         x=c(aresv1, aresv2, kresv1, kresv2),
                       legend=rep(c("aresv1", "aresv2", "kresv1", "kresv2"),
                       c(len(aresv1), len(aresv2), len(kresv1), len(kresv2))))
  pdat_cuts <- data.frame(names=c(paste0("mean", 1:2), paste0("med", 1:2)), vals=c(means, meds))
  pdat_pt <- data.frame(val=param_true,y=0)

  p <- ggplot()
  p <- p + geom_hline(data=pdat_pt, aes(xintercept=y), col="grey", size=1)
  p <- p + labs(list(x=labx, y=expression(paste("LINEX parameter ", bolditalic(a),
       " / General Entropy parameter  ", bolditalic(k)))))

  p <- p + geom_vline(data=pdat_cuts, aes(xintercept=vals, col=names), size=1.2, show_guide=T)
  p <- p + geom_line(data=pdat, aes(x=x, y=y, col=legend), size=1.1) +
       geom_point(data=pdat, aes(x=x, y=y, col=legend), size=3)
  p <- p + geom_vline(data=pdat_pt, aes(xintercept=val), col="black", linetype="longdash", size=2)

  number_ticks <- function(n) {function(limits) pretty(limits, n)}
  p <- p + scale_x_continuous(breaks=number_ticks(x_nbreaks))
  if(!missing(limx))
    p <- p + coord_cartesian(xlim=limx)
  if(!missing(limy))
    p <- p + ylim(limy)

  p2 <- p + scale_colour_manual(values=c("gold2", "orange", "violet", "red", "steelblue2", "blue", "green",
       "forestgreen"), labels=c(paste("LNX loss ests.", c("(0% cens.)", "(20% cens.)")),
       paste("GENT loss ests.", c("(0% cens.)", "(20% cens.)")),
       paste("SE loss est.", c("(0% cens.)", "(20% cens.)")),
       paste("AE loss est.", c("(0% cens.)", "(20% cens.)"))))
  p2 <- p2 + theme(legend.title=element_blank())

  if(show_plot){
    if(win_plot)
      windows(width, height)
    print(p2)

    if(!missing(fname))
      ggsave(file=fname)
  }

  return(invisible(p2))
}
```

```
75
76
77  bayes_est_ggplot_prior <- function(sim1_jeff, sim1_ref1, sim1_ref2, sim2_jeff, sim2_ref1, sim2_ref2,
78                                     model, param_true, fname, x_nbreaks=10, limx, limy,
79                                     show_plot=TRUE, win_plot=TRUE, width=14, height=6)
80  {
81    if(model == "crgA"){
82      labx   <- expression(paste(alpha, " parameter value"))
83      a_ind <- 7 ; p_ind <- 3 ; k_ind <- 9
84    }
85    else if(model == "crgB"){
86      labx   <- expression(paste(beta, " parameter value"))
87      a_ind <- 8 ; p_ind <- 4 ; k_ind <- 10
88    }
89    else if(model == "gcrgA"){
90      labx   <- expression(paste(alpha, " parameter value"))
91      a_ind <- 10 ; p_ind <- 4 ; k_ind <- 13
92    }
93    else if(model == "gcrgB"){
94      labx   <- expression(paste(beta, " parameter value"))
95      a_ind <- 11 ; p_ind <- 5 ; k_ind <- 14
96    }
97    else if(model == "gcrgC"){
98      labx   <- expression(paste(italic(c), " parameter value"))
99      a_ind <- 12 ; p_ind <- 6 ; k_ind <- 15
100   }
101   else
102     stop("Details incorrectly specified.")
103
104   ## 0% censoring ##
105   aresv1_p1  <- sim1_jeff[[a_ind]]
106   aresv1_p2  <- sim1_ref1[[a_ind]]
107   aresv1_p3  <- sim1_ref2[[a_ind]]
108   kresv1_p1  <- sim1_jeff[[k_ind]]
109   kresv1_p2  <- sim1_ref1[[k_ind]]
110   kresv1_p3  <- sim1_ref2[[k_ind]]
111   pstats1_p1 <- sim1_jeff[[p_ind]]
112   pstats1_p2 <- sim1_ref1[[p_ind]]
113   pstats1_p3 <- sim1_ref2[[p_ind]]
114
115   means1 <- c(pstats1_p1[3], pstats1_p2[3], pstats1_p3[3])
116   meds1  <- c(pstats1_p1[2], pstats1_p2[2], pstats1_p3[2])
117
118   ## 20% censoring ##
119   aresv2_p1  <- sim2_jeff[[a_ind]]
120   aresv2_p2  <- sim2_ref1[[a_ind]]
121   aresv2_p3  <- sim2_ref2[[a_ind]]
122   kresv2_p1  <- sim2_jeff[[k_ind]]
123   kresv2_p2  <- sim2_ref1[[k_ind]]
124   kresv2_p3  <- sim2_ref2[[k_ind]]
125   pstats2_p1 <- sim2_jeff[[p_ind]]
126   pstats2_p2 <- sim2_ref1[[p_ind]]
127   pstats2_p3 <- sim2_ref2[[p_ind]]
128
129   means2 <- c(pstats2_p1[3], pstats2_p2[3], pstats2_p3[3])
130   meds2  <- c(pstats2_p1[2], pstats2_p2[2], pstats2_p3[2])
131
132   len <- function(...) length(...)
133   nn  <- function(...) as.numeric(names(...))
134   pdat <- data.frame(y=c(nn(aresv1_p1), nn(aresv1_p2), nn(aresv1_p3), nn(kresv1_p1), nn(kresv1_p2),
135                          nn(kresv1_p3), nn(aresv2_p1), nn(aresv2_p2), nn(aresv2_p2), nn(kresv2_p1),
136                          nn(kresv2_p2), nn(kresv2_p2)), x=c(aresv1_p1, aresv1_p2, aresv1_p3,
137                          kresv1_p1, kresv1_p2, kresv1_p3, aresv2_p1, aresv2_p2, aresv2_p3,
138                          kresv2_p1, kresv2_p2, kresv2_p3), legend=rep(c("ares_p1", "ares_p2", "ares_p3",
139                          "kres_p1", "kres_p2", "kres_p3", "ares_p1", "ares_p2", "ares_p3",
```

```
                        "kres_p1", "kres_p2", "kres_p3"), c(len(aresv1_p1), len(aresv1_p2),
                        len(aresv1_p3), len(kresv1_p1), len(kresv1_p2), len(kresv1_p3),
                        len(aresv2_p1), len(aresv2_p2), len(aresv2_p3), len(kresv2_p1),
                        len(kresv2_p2), len(kresv2_p3))), cens_level=rep(c("cens1", "cens2"),
                        c(len(aresv1_p1) + len(aresv1_p2) + len(aresv1_p3) + len(kresv1_p1) +
                        len(kresv1_p2) + len(kresv1_p3), len(aresv2_p1) + len(aresv2_p2) +
                        len(aresv2_p3) + len(kresv2_p1) + len(kresv2_p2) + len(kresv2_p3))))

  pdat_cuts <- data.frame(names=c(paste0("mean", 1:3), paste0("med", 1:3), paste0("mean", 1:3),
                          paste0("med", 1:3)), vals=c(means1, meds1, means2, meds2),
                          cens_level=rep(c("cens1", "cens2"), c(6,6)))
  pdat_pt <- data.frame(val=param_true)

  p <- ggplot() + labs(list(x=labx, y=expression(paste("LINEX parameter ", bolditalic(a),
       " / General Entropy parameter  ", bolditalic(k)))))
  p <- p + geom_hline(data=pdat_pt, aes(xintercept=0), col="grey", size=1)
  p <- p + geom_vline(data=pdat_cuts, aes(xintercept=vals, col=names), size=1.2, show_guide=T)
  p <- p + geom_line(data=pdat, aes(x=x, y=y, col=legend), size=1.1) +
       geom_point(data=pdat, aes(x=x, y=y, col=legend), size=3)
  p <- p + geom_vline(data=pdat_pt, aes(xintercept=val), col="black", linetype="longdash", size=2)

  ## faceting ##
  facet_names <- list("cens1"="0% censored", "cens2"="20% censored")
  facet_label_fun <- function(variable, value)
    return(facet_names[value])
  p <- p + facet_grid(cens_level~., labeller=facet_label_fun)

  number_ticks <- function(n) {function(limits) pretty(limits, n)}
  p <- p + scale_x_continuous(breaks=number_ticks(x_nbreaks))
  if(!missing(limx))
    p <- p + coord_cartesian(xlim=limx)
  if(!missing(limy))
    p <- p + ylim(limy)

  p2 <- p + scale_colour_manual(values=c("yellow", "gold2", "orange", "violet", "maroon", "red",
                                         "lightslateblue", "blue", "steelblue4", "chartreuse", "limegreen",
                                         "forestgreen"),
                                labels=c(paste("LNX loss", c("(Jeffreys)", "(reference 1)",
                                         "(reference 2)")), paste("GENT loss", c("(Jeffreys)",
                                         "(reference 1)", "(reference 2)")), paste("SE loss",
                                         c("(Jeffreys)", "(reference 1)", "(reference 2)")),
                                         paste("AE loss", c("(Jeffreys)", "(reference 1)",
                                         "(reference 2)"))))
  p2 <- p2 + theme(legend.title=element_blank())

  if(show_plot){
    if(win_plot)
      windows(width, height)
    print(p2)

    if(!missing(fname))
      ggsave(file=fname)
  }

  return(invisible(p2))
}


bayes_est_funs_ggplot_cens <- function(mat1, mat2, tlim, show_plot=TRUE, win_plot=TRUE,
                                       width=14, height=6, fname)
{
  len <- function(...) length(...)

  aExtract <- function(emat){
    arows <- (1:len(rownames(emat)))[sapply(rownames(emat), function(x) substring(x, 1, 1)) == "L"]
```

```
  return(as.numeric(sapply(rownames(emat)[arows], function(x) as.numeric(substring(x,9,nchar(x))))) )
}

kExtract <- function(emat){
  krows <- (1:len(rownames(emat)))[sapply(rownames(emat), function(x) substring(x, 1, 1)) == "G"]
  return(as.numeric(sapply(rownames(emat)[krows], function(x) as.numeric(substring(x,8,nchar(x))))) )
}

ttExtract <- function(emat){
  colnum <- as.numeric(colnames(emat))
  return( seq(colnum[1], colnum[length(colnum)], len=length(colnum)) )
}

pdatExtract <- function(em1, em2, vec, i_start){
  pd <- data.frame(y=NULL, x=NULL, legend=NULL)
  for(i in (i_start):(i_start+len(vec)-1))
    pd <- rbind(pd, data.frame(y=c(em1[i,], em2[i,]), x=rep(tt,2),
                               legend=rep(paste0(paste0("row",i), c("cens1", "cens2")), each=len(tt))))
  return(pd)
}

if(!all(dim(mat1) == dim(mat2)))
  stop("Input matrices does not have equal dimensions.")

if(!all(mat1[1,] == mat2[1,]))
  stop("True values not consistent across input matrices.")

avec <- aExtract(mat1)
kvec <- kExtract(mat1)
tt <- ttExtract(mat1)
f_true <- as.numeric(mat1[1,])
emat1 <- mat1[-1,]
emat2 <- mat2[-1,]

pdat_aese <- data.frame(y=c(emat1[1,], emat2[1,], emat1[2,], emat2[2,]),
                        x=rep(tt, 4), legend=rep(c(paste0("row1", c("cens1", "cens2")),
                        paste0("row2", c("cens1", "cens2"))), each=len(tt)))
pdat_avec <- pdat_avec <- pdatExtract(emat1, emat2, avec, 3)
pdat_kvec <- pdat_kvec <- pdatExtract(emat1, emat2, kvec, 3+len(avec))

pdat <- cbind(rbind(pdat_aese, pdat_avec, pdat_kvec), facet_lvls=rep(paste0("plot", 1:3),
              c(nrow(pdat_aese), nrow(pdat_avec), nrow(pdat_kvec))))

pdat_ax   <- data.frame(y=0)
pdat_true <- data.frame(y=rep(f_true,3), x=rep(tt,3), legend=rep("f_true", 3*len(tt)))

p <- ggplot() + geom_hline(data=pdat_ax, aes(xintercept=y), col="grey", size=1)
p <- p + labs(list(x="time", y="hazard rate"))

p <- p + geom_line(data=pdat, aes(x=x, y=y, col=legend, linetype=legend), size=1.1)
p <- p + geom_line(data=pdat_true, aes(x=x, y=y, col=legend, linetype=legend), size=1.3)

if(!missing(tlim))
  p <- p + coord_cartesian(xlim=tlim)

## faceting ##
facet_names <- list("plot1"="Abs and SE loss estimates", "plot2"="LINEX(a) loss estimates",
                    "plot3"="GenEnt(k) loss estimates")
facet_label_fun <- function(variable, value)
  return(facet_names[value])
p <- p + facet_grid(facet_lvls~., labeller=facet_label_fun)

avec_names <- c(sapply(paste0("LNX(", avec, ") loss"),
                function(x) paste(x, c("(0% cens.)", "(20% cens.)"))))
kvec_names <- c(sapply(paste0("GENT(", kvec, ") loss"),
```

```
                  function(x) paste(x, c("(0% cens.)", "(20% cens.)"))))
  p2 <- p + scale_colour_manual(values=c("black", "blue", "blue", "green", "green",
                                rep(c("yellow", "orange"), each=2), rep(c("violet", "red"), each=2)),
                                labels=c("true hazard", paste("SE loss", c("(0% cens.)", "(20% cens.)")),
                                paste("AE loss", c("(0% cens.)", "(20% cens.)")), avec_names, kvec_names))
  p2 <- p2 + theme(legend.title=element_blank())

  p2 <- p2 + scale_linetype_manual(values=c("longdash", rep(c("solid", "dotdash"),6)),
                                   labels=c("true hazard", paste("SE loss", c("(0% cens.)", "(20% cens.)")),
                                   paste("AE loss", c("(0% cens.)", "(20% cens.)")), avec_names, kvec_names))

  if(show_plot){
    if(win_plot)
      windows(width, height)
    print(p2)

    if(!missing(fname))
      ggsave(file=fname)
  }

  return(invisible(p2))
}


bayes_est_funs_ggplot_cens <- function(mat1, mat2, tlim, show_plot=TRUE, win_plot=TRUE,
                                       width=14, height=6, fname)
{
  len <- function(...) length(...)

  aExtract <- function(emat){
    arows <- (1:len(rownames(emat)))[sapply(rownames(emat), function(x) substring(x, 1, 1)) == "L"]
    return(as.numeric(sapply(rownames(emat)[arows], function(x) as.numeric(substring(x,9,nchar(x))))) )
  }

  kExtract <- function(emat){
    krows <- (1:len(rownames(emat)))[sapply(rownames(emat), function(x) substring(x, 1, 1)) == "G"]
    return(as.numeric(sapply(rownames(emat)[krows], function(x) as.numeric(substring(x,8,nchar(x))))) )
  }

  ttExtract <- function(emat){
    colnum <- as.numeric(colnames(emat))
    return( seq(colnum[1], colnum[length(colnum)], len=length(colnum)) )
  }

  pdatExtract <- function(em1, em2, vec, i_start){
    pd <- data.frame(y=NULL, x=NULL, legend=NULL)
    for(i in (i_start):(i_start+len(vec)-1))
      pd <- rbind(pd, data.frame(y=c(em1[i,], em2[i,]), x=rep(tt,2),
                                 legend=rep(paste0(paste0("row",i), c("cens1", "cens2")), each=len(tt))))
    return(pd)
  }

  if(!all(dim(mat1) == dim(mat2)))
    stop("Input matrices does not have equal dimensions.")

  if(!all(mat1[1,] == mat2[1,]))
    stop("True values not consistent across input matrices.")

  avec1 <- aExtract(mat1)
  kvec1 <- kExtract(mat1)
  avec2 <- aExtract(mat2)
  kvec2 <- kExtract(mat2)

  tt <- ttExtract(mat1)
  f_true <- as.numeric(mat1[1,])
```

```
  emat1 <- mat1[-1,]
  emat2 <- mat2[-1,]

  pdat_aese <- data.frame(y=c(emat1[1,], emat2[1,], emat1[2,], emat2[2,]),
                          x=rep(tt, 4), legend=rep(c(paste0("row1", c("cens1", "cens2")),
                          paste0("row2", c("cens1", "cens2"))), each=len(tt)))
  pdat_avec <- pdat_avec <- pdatExtract(emat1, emat2, avec1, 3)
  pdat_kvec <- pdat_kvec <- pdatExtract(emat1, emat2, kvec1, 3+len(avec1))

  pdat <- cbind(rbind(pdat_aese, pdat_avec, pdat_kvec), facet_lvls=rep(paste0("plot", c(1,2,2)),
               c(nrow(pdat_aese), nrow(pdat_avec), nrow(pdat_kvec))))

  pdat_ax   <- data.frame(y=0)
  pdat_true <- data.frame(y=rep(f_true,3), x=rep(tt,3), legend=rep("f_true", 3*len(tt)))

  p <- ggplot() + geom_hline(data=pdat_ax, aes(xintercept=y), col="grey", size=1)
  p <- p + labs(list(x="time", y="hazard rate"))

  p <- p + geom_line(data=pdat, aes(x=x, y=y, col=legend, linetype=legend), size=1.1)
  p <- p + geom_line(data=pdat_true, aes(x=x, y=y, col=legend, linetype=legend), size=1.3)

  if(!missing(tlim))
    p <- p + coord_cartesian(xlim=tlim)

  ## faceting ##
  facet_names <- list("plot1"="AE and SE loss estimates",
                      "plot2"="LNX(a) and GENT(k) estimates")
  facet_label_fun <- function(variable, value)
    return(facet_names[value])
  p <- p + facet_grid(facet_lvls~., labeller=facet_label_fun)

  avec_names <- paste(c(paste0("LNX(", avec1, ") loss"),
                      paste0("LNX(", avec2, ") loss")), c("(0% cens.)", "(20% cens.)"))
  kvec_names <- paste(c(paste0("GENT(", kvec1, ") loss"),
                      paste0("GENT(", kvec2, ") loss")), c("(0% cens.)", "(20% cens.)"))

  p2 <- p + scale_colour_manual(values=c("black", "blue", "blue", "green", "green",
                                rep(c("red", "orange"), each=2), rep(c("violet", "yellow"), each=2)),
                                labels=c("true hazard", paste("SE loss", c("(0% cens.)", "(20% cens.)")),
                                paste("AE loss", c("(0% cens.)", "(20% cens.)")), avec_names, kvec_names))
  p2 <- p2 + theme(legend.title=element_blank())

  p2 <- p2 + scale_linetype_manual(values=c("longdash", rep(c("solid", "dotdash"),6)),
        labels=c("true hazard", paste("SE loss", c("(0% cens.)", "(20% cens.)")),
        paste("AE loss", c("(0% cens.)", "(20% cens.)")), avec_names, kvec_names))

  if(show_plot){
    if(win_plot)
      windows(width, height)
    print(p2)

    if(!missing(fname))
      ggsave(file=fname)
  }

  return(invisible(p2))
}


bayes_est_funs_ggplot_prior <- function(mat1, mat2, mat3, tlim,
                                        show_plot=TRUE, win_plot=TRUE, width=14, height=6, fname)
{
  len <- function(...) length(...)

  aExtract <- function(emat){
```

```
  arows <- (1:len(rownames(emat)))[sapply(rownames(emat), function(x) substring(x, 1, 1)) == "L"]
  return(as.numeric(sapply(rownames(emat)[arows], function(x) as.numeric(substring(x,9,nchar(x))))) )
}

kExtract <- function(emat){
  krows <- (1:len(rownames(emat)))[sapply(rownames(emat), function(x) substring(x, 1, 1)) == "G"]
  return(as.numeric(sapply(rownames(emat)[krows], function(x) as.numeric(substring(x,8,nchar(x))))) )
}

ttExtract <- function(emat){
  colnum <- as.numeric(colnames(emat))
  return( seq(colnum[1], colnum[length(colnum)], len=length(colnum)) )
}

pdatExtract <- function(em1, em2, em3, vec, i_start){
  pd <- data.frame(y=NULL, x=NULL, legend=NULL)
  for(i in (i_start):(i_start+len(vec)-1))
    pd <- rbind(pd, data.frame(y=c(em1[i,], em2[i,], em3[i,]), x=rep(tt,3),
                               legend=rep(paste0(paste0("row",i), c("pr1", "pr2", "pr3")), each=len(tt))))
  return(pd)
}

if(!(all(dim(mat1) == dim(mat2)) & all(dim(mat1) == dim(mat3))))
  stop("Input matrices does not have equal dimensions.")

if(!(all(mat1[1,] == mat2[1,]) & all(mat1[1,] == mat3[1,])))
  stop("True values not consistent across input matrices.")

avec   <- aExtract(mat1)
kvec   <- kExtract(mat1)
tt     <- ttExtract(mat1)
f_true <- as.numeric(mat1[1,])
emat1  <- mat1[-1,]
emat2  <- mat2[-1,]
emat3  <- mat3[-1,]

pdat_aese <- data.frame(y=c(emat1[1,], emat2[1,], emat3[1,], emat1[2,], emat2[2,], emat3[2,]),
                        x=rep(tt, 6), legend=rep(c(paste0("row1", c("pr1", "pr2", "pr3")),
                        paste0("row2", c("pr1", "pr2", "pr3"))), each=len(tt)))
pdat_avec <- pdatExtract(emat1, emat2, emat3, avec, 3)
pdat_kvec <- pdatExtract(emat1, emat2, emat3, kvec, 3+len(avec))

pdat <- cbind(rbind(pdat_aese, pdat_avec, pdat_kvec), facet_lvls=rep(paste0("plot", 1:3),
              c(nrow(pdat_aese), nrow(pdat_avec), nrow(pdat_kvec))))

pdat_ax   <- data.frame(y=0)
pdat_true <- data.frame(y=rep(f_true,3), x=rep(tt,3), legend=rep("f_true", 3*len(tt)))

p <- ggplot() + geom_hline(data=pdat_ax, aes(xintercept=y), col="grey", size=1)
p <- p + labs(list(x="time", y="hazard rate"))

p <- p + geom_line(data=pdat, aes(x=x, y=y, col=legend, linetype=legend), size=1.1)
p <- p + geom_line(data=pdat_true, aes(x=x, y=y, col=legend, linetype=legend), size=1.3)

if(!missing(tlim))
  p <- p + coord_cartesian(xlim=tlim)

## faceting ##
facet_names <- list("plot1"="Abs and SE loss estimates", "plot2"="LINEX(a) loss estimates",
                    "plot3"="GenEnt(k) loss estimates")
facet_label_fun <- function(variable, value)
  return(facet_names[value])
p <- p + facet_grid(facet_lvls~., labeller=facet_label_fun)

avec_names <- c(sapply(paste0("LNX(", avec, ") loss"),
```

```
                  function(x) paste(x, c("(Jeffreys)", "(Reference)", "(P. Matching)"))))
  kvec_names <- c(sapply(paste0("GENT(", kvec, ") loss"),
                  function(x) paste(x, c("(Jeffreys)", "(Reference)", "(P. Matching)"))))

  p2 <- p + scale_colour_manual(values=c("black", "blue", "blue", "blue", "green", "green", "green",
                                rep(c("yellow", "orange"), each=3), rep(c("violet", "red"), each=3)),
                                labels=c("true hazard", paste("SE loss",
                                c("(Jeffreys)", "(Reference)", "(P. Matching)")),
                                paste("AE loss", c("(Jeffreys)", "(Reference)", "(P. Matching)")),
                                avec_names, kvec_names))
  p2 <- p2 + theme(legend.title=element_blank())

  p2 <- p2 + scale_linetype_manual(values=c("longdash", rep(c("solid", "dotdash", "dashed"), 6)),
                                   labels=c("true hazard", paste("SE loss", c("(Jeffreys)", "(Reference)",
                                   "(P. Matching)")), paste("AE loss", c("(Jeffreys)", "(Reference)",
                                   "(P. Matching)")), avec_names, kvec_names))

  if(show_plot){
    if(win_plot)
      windows(width, height)
    print(p2)

    if(!missing(fname))
      ggsave(file=fname)
  }

  return(invisible(p2))
}


bayes_est_funs_ggplot_prior <- function(mat1, mat2, mat3, tlim,
                                        show_plot=TRUE, win_plot=TRUE, width=14, height=6, fname)
{
  len <- function(...) length(...)

  aExtract <- function(emat){
    arows <- (1:len(rownames(emat)))[sapply(rownames(emat), function(x) substring(x, 1, 1)) == "L"]
    return(as.numeric(sapply(rownames(emat)[arows], function(x) as.numeric(substring(x,9,nchar(x))))) )
  }

  kExtract <- function(emat){
    krows <- (1:len(rownames(emat)))[sapply(rownames(emat), function(x) substring(x, 1, 1)) == "G"]
    return(as.numeric(sapply(rownames(emat)[krows], function(x) as.numeric(substring(x,8,nchar(x))))) )
  }

  ttExtract <- function(emat){
    colnum <- as.numeric(colnames(emat))
    return( seq(colnum[1], colnum[length(colnum)], len=length(colnum)) )
  }

  pdatExtract <- function(em1, em2, em3, vec, i_start){
    pd <- data.frame(y=NULL, x=NULL, legend=NULL)
    for(i in (i_start):(i_start+len(vec)-1))
      pd <- rbind(pd, data.frame(y=c(em1[i,], em2[i,], em3[i,]), x=rep(tt,3),
                                 legend=rep(paste0(paste0("row",i), c("pr1", "pr2", "pr3")), each=len(tt))))
    return(pd)
  }

  if(!(all(dim(mat1) == dim(mat2)) & all(dim(mat1) == dim(mat3))))
    stop("Input matrices does not have equal dimensions.")

  if(!(all(mat1[1,] == mat2[1,]) & all(mat1[1,] == mat3[1,])))
    stop("True values not consistent across input matrices.")

  avec1   <- aExtract(mat1)
```

```
avec2   <- aExtract(mat2)
avec3   <- aExtract(mat3)
kvec1   <- kExtract(mat1)
kvec2   <- kExtract(mat2)
kvec3   <- kExtract(mat3)

tt     <- ttExtract(mat1)
f_true <- as.numeric(mat1[1,])
emat1  <- mat1[-1,]
emat2  <- mat2[-1,]
emat3  <- mat3[-1,]

pdat_aese <- data.frame(y=c(emat1[1,], emat2[1,], emat3[1,], emat1[2,], emat2[2,], emat3[2,]),
                        x=rep(tt, 6), legend=rep(c(paste0("row1", c("pr1", "pr2", "pr3")),
                        paste0("row2", c("pr1", "pr2", "pr3"))), each=len(tt)))
pdat_avec <- pdatExtract(emat1, emat2, emat3, avec1, 3)
pdat_kvec <- pdatExtract(emat1, emat2, emat3, kvec1, 3+len(avec1))

pdat <- cbind(rbind(pdat_aese, pdat_avec, pdat_kvec), facet_lvls=rep(paste0("plot", c(1,2,2)),
              c(nrow(pdat_aese), nrow(pdat_avec), nrow(pdat_kvec))))

pdat_ax   <- data.frame(y=0)
pdat_true <- data.frame(y=rep(f_true,3), x=rep(tt,3), legend=rep("f_true", 3*len(tt)))

p <- ggplot() + geom_hline(data=pdat_ax, aes(xintercept=y), col="grey", size=1)
p <- p + labs(list(x="time", y="hazard rate"))

p <- p + geom_line(data=pdat, aes(x=x, y=y, col=legend, linetype=legend), size=1.1)
p <- p + geom_line(data=pdat_true, aes(x=x, y=y, col=legend, linetype=legend), size=1.3)

if(!missing(tlim))
  p <- p + coord_cartesian(xlim=tlim)

## faceting ##
facet_names <- list("plot1"="AE and SE loss estimates", "plot2"="LNX(a) and GENT(k) estimates")
facet_label_fun <- function(variable, value)
  return(facet_names[value])
p <- p + facet_grid(facet_lvls~., labeller=facet_label_fun)

avec_names <- c(paste0("LNX(", avec1,") loss (Jeffreys)"),
                paste0("LNX(", avec2,") loss (Reference)"),
                paste0("LNX(", avec3,") loss (P. Matching)"))
kvec_names <- c(paste0("GENT(", kvec1,") loss (Jeffreys)"),
                paste0("GENT(", kvec2,") loss (Reference)"),
                paste0("GENT(", kvec3,") loss (P. Matching)"))

p2 <- p + scale_colour_manual(values=c("black", "blue", "blue", "blue", "green", "green", "green",
                              rep(c("red", "orange"), each=3), rep(c("violet", "yellow"), each=3)),
                              labels=c("true hazard", paste("SE loss", c("(Jeffreys)", "(Reference)",
                              "(P. Matching)")), paste("AE loss", c("(Jeffreys)", "(Reference)",
                              "(P. Matching)")), avec_names, kvec_names))
p2 <- p2 + theme(legend.title=element_blank())

p2 <- p2 + scale_linetype_manual(values=c("longdash", rep(c("solid", "dotdash", "dashed"), 6)),
      labels=c("true hazard", paste("SE loss", c("(Jeffreys)", "(Reference)", "(P. Matching)")),
      paste("AE loss", c("(Jeffreys)", "(Reference)", "(P. Matching)")), avec_names, kvec_names))

if(show_plot){
  if(win_plot)
    windows(width, height)
  print(p2)

  if(!missing(fname))
    ggsave(file=fname)
}
```

```
595
596   return(invisible(p2))
597 }
598
599
600 MB_ggplot_cens <- function(mb1, mb2, param_true, model,
601                            show_plot=TRUE, win_plot=TRUE, width=14, height=6, fname)
602 {
603   if(model == "cre"){
604     labx   <- expression(paste(gamma, " parameter value"))
605     p_ind <- 3
606   }
607   else if(model == "crgA"){
608     labx   <- expression(paste(alpha, " parameter value"))
609     p_ind <- 5
610   }
611   else if(model == "crgB"){
612     labx   <- expression(paste(beta, " parameter value"))
613     p_ind <- 6
614   }
615   else if(model == "gcreG"){
616     labx   <- expression(paste(gamma, " parameter value"))
617     p_ind <- 5
618   }
619   else if(model == "gcreC"){
620     labx   <- expression(paste(italic(c), " parameter value"))
621     p_ind <- 6
622   }
623   else if(model == "gcrgA"){
624     labx   <- expression(paste(alpha, " parameter value"))
625     p_ind <- 7
626   }
627   else if(model == "gcrgB"){
628     labx   <- expression(paste(beta, " parameter value"))
629     p_ind <- 8
630   }
631   else if(model == "gcrgC"){
632     labx   <- expression(paste(italic(c), " parameter value"))
633     p_ind <- 9
634   }
635   else
636     stop("Details incorrectly specified.")
637
638   params1 <- as.numeric(colnames(mb1))
639   params2 <- as.numeric(colnames(mb2))
640   col_true1 <- which(params1 == param_true)
641   col_true2 <- which(params2 == param_true)
642
643
644   pdat <- data.frame(y=c(mb1[1,], mb2[1,]), x=c(mb1[2,], mb2[2,]),
645                      legend=rep(c("cens1", "cens2"), c(ncol(mb1), ncol(mb2))))
646
647   pdat_txt <- data.frame(y=c(mb1[1,], mb2[1,]), x=c(mb1[2,], mb2[2,]),
648                          txt_labs=c(as.character(params1), as.character(params2)))
649
650   pdat_true <- data.frame(y=c(mb1[1,col_true1], mb2[1,col_true2]),
651                           x=c(mb1[2,col_true1], mb2[2,col_true2]))
652
653   p <- ggplot() + labs(list(x="bias", y="mean squared error"))
654
655   p <- p + geom_path(data=pdat, aes(x=x, y=y, col=legend), size=1) +
656        geom_point(data=pdat, aes(x=x, y=y, col=legend), size=3)
657
658   p <- p + geom_point(data=pdat_true, aes(x=x, y=y), col="red", size=4)
659
```

```
  p <- p + geom_text(data=pdat_txt, aes(x=x, y=y, label=txt_labs), hjust=0, vjust=1.25, size=3.5)

  p2 <- p + scale_colour_manual(values=c("steelblue", "blue"), labels=c("0% censoring", "20% censoring"))
  p2 <- p2 + theme(legend.title=element_blank())

  if(show_plot){
    if(win_plot)
      windows(width, height)
    print(p2)

    if(!missing(fname))
      ggsave(file=fname)
  }

  return(invisible(p2))
}


MB_ggplot_prior <- function(mb1, mb2, mb3, param_true, model,
                            show_plot=TRUE, win_plot=TRUE, width=14, height=6, fname)
{
  if(model == "crgA"){
    labx   <- expression(paste(alpha, " parameter value"))
    p_ind <- 5
  }
  else if(model == "crgB"){
    labx   <- expression(paste(beta, " parameter value"))
    p_ind <- 6
  }
  else if(model == "gcrgA"){
    labx   <- expression(paste(alpha, " parameter value"))
    p_ind <- 7
  }
  else if(model == "gcrgB"){
    labx   <- expression(paste(beta, " parameter value"))
    p_ind <- 8
  }
  else if(model == "gcrgC"){
    labx   <- expression(paste(italic(c), " parameter value"))
    p_ind <- 9
  }
  else
    stop("Details incorrectly specified.")

  params1 <- as.numeric(colnames(mb1))
  params2 <- as.numeric(colnames(mb2))
  params3 <- as.numeric(colnames(mb3))
  col_true1 <- which(params1 == param_true)
  col_true2 <- which(params2 == param_true)
  col_true3 <- which(params3 == param_true)

  pdat <- data.frame(y=c(mb1[1,], mb2[1,], mb3[1,]), x=c(mb1[2,], mb2[2,], mb3[2,]),
                     legend=rep(c("prior1", "prior2", "prior3"), c(ncol(mb1), ncol(mb2), ncol(mb3))))

  pdat_txt <- data.frame(y=c(mb1[1,], mb2[1,], mb3[1,]), x=c(mb1[2,], mb2[2,], mb3[2,]),
                         txt_labs=c(as.character(params1), as.character(params2), as.character(params3)))

  pdat_true <- data.frame(y=c(mb1[1,col_true1], mb2[1,col_true2], mb3[1,col_true3]),
                          x=c(mb1[2,col_true1], mb2[2,col_true2], mb3[2,col_true3]))

  p <- ggplot() + labs(list(x="bias", y="mean squared error"))

  p <- p + geom_path(data=pdat, aes(x=x, y=y, col=legend), size=1) +
       geom_point(data=pdat, aes(x=x, y=y, col=legend), size=3)

```

```
  p <- p + geom_point(data=pdat_true, aes(x=x, y=y), col="red", size=4)

  p <- p + geom_text(data=pdat_txt, aes(x=x, y=y, label=txt_labs), hjust=0, vjust=1.25, size=3.5)

  p2 <- p + scale_colour_manual(values=c("steelblue", "blue", "darkblue"), labels=c("Jeffreys", "Reference",
      "PM"))
  p2 <- p2 + theme(legend.title=element_blank())

  if(show_plot){
    if(win_plot)
      windows(width, height)
    print(p2)

    if(!missing(fname))
      ggsave(file=fname)
  }

  return(invisible(p2))
}


cov_ggplot_cens_univar <- function(sim1, sim2, param_true, model, limx, x_nbreaks=6,
                                   show_plot=TRUE, win_plot=TRUE, width=14, height=6, fname)
{
  if(model == "cre"){
    labx   <- expression(paste(gamma, " parameter value"))
    p_ind <- 3
  }
  else if(model == "crgA"){
    labx   <- expression(paste(alpha, " parameter value"))
    p_ind <- 5
  }
  else if(model == "crgB"){
    labx   <- expression(paste(beta, " parameter value"))
    p_ind <- 6
  }
  else if(model == "gcreG"){
    labx   <- expression(paste(gamma, " parameter value"))
    p_ind <- 5
  }
  else if(model == "gcreC"){
    labx   <- expression(paste(italic(c), " parameter value"))
    p_ind <- 6
  }
  else if(model == "gcrgA"){
    labx   <- expression(paste(alpha, " parameter value"))
    p_ind <- 7
  }
  else if(model == "gcrgB"){
    labx   <- expression(paste(beta, " parameter value"))
    p_ind <- 8
  }
  else if(model == "gcrgC"){
    labx   <- expression(paste(italic(c), " parameter value"))
    p_ind <- 9
  }
  else
    stop("Details incorrectly specified.")

  parstats1 <- sim1[[p_ind]]
  parstats2 <- sim2[[p_ind]]

  cov1  <- parstats1[3,]*100
  cov2  <- parstats2[3,]*100
  params1 <- as.numeric(colnames(parstats1))
```

```
  params2 <- as.numeric(colnames(parstats2))
  col_true1 <- which(params1 == param_true)
  col_true2 <- which(params2 == param_true)

  pdat <- data.frame(y=c(cov1, cov2), x=c(params1, params2),
                     legend=rep(c("cens1", "cens2"), c(length(cov1), length(cov2))))
  pdat_lines <- data.frame(x=param_true, y=95)

  p <- ggplot() + labs(list(x=labx, y="coverage (%)"))

  p <- p + geom_vline(data=pdat_lines, aes(xintercept=x), col="red", linetype="longdash", size=2)
  p <- p + geom_hline(data=pdat_lines, aes(yintercept=y), linetype="dashed", size=1)

  p <- p + geom_path(data=pdat, aes(x=x, y=y, col=legend), size=1) +
    geom_point(data=pdat, aes(x=x, y=y, col=legend), size=3)

  p <- p + scale_y_continuous(breaks=100*c(0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1))
  p <- p + coord_cartesian(ylim=100*c(0.4,1.01))

  number_ticks <- function(n) {function(limits) pretty(limits, n)}
  if(!missing(limx))
    p <- p + scale_x_continuous(limits=limx, breaks=number_ticks(x_nbreaks))
  else
    p <- p + scale_x_continuous(breaks=number_ticks(x_nbreaks))

  p2 <- p + scale_colour_manual(values=c("green", "mediumseagreen"),
                                labels=c("0% censoring", "20% censoring"))
  p2 <- p2 + theme(legend.title=element_blank())

  if(show_plot){
    if(win_plot)
      windows(width, height)
    print(p2)

    if(!missing(fname))
      ggsave(file=fname)
  }

  return(invisible(p2))
}


cov_ggplot_cens_bivar <- function(sim1, sim2, param_true1, param_true2, model, limx, x_nbreaks=6,
                                  show_plot=TRUE, win_plot=TRUE, width=14, height=6, fname)
{
  if(model == "crg"){
    labx1   <- expression(paste(alpha, " parameter"))
    labx2   <- expression(paste(beta, " parameter"))
  }
  else if(model == "gcre"){
    labx1   <- expression(paste(gamma, " parameter"))
    labx2   <- expression(paste(italic(c), " parameter"))
  }
  else
    stop("Details incorrectly specified.")

  p1_ind <- 5
  p2_ind <- 6

  parstats1_1 <- sim1[[p1_ind]]
  parstats2_1 <- sim2[[p1_ind]]
  parstats1_2 <- sim1[[p2_ind]]
  parstats2_2 <- sim2[[p2_ind]]

  cov1_1  <- parstats1_1[3,]*100
```

```
  cov1_2  <- parstats1_2[3,]*100
  cov2_1  <- parstats2_1[3,]*100
  cov2_2  <- parstats2_2[3,]*100
  params1_1 <- as.numeric(colnames(parstats1_1))
  params1_2 <- as.numeric(colnames(parstats1_2))
  params2_1 <- as.numeric(colnames(parstats2_1))
  params2_2 <- as.numeric(colnames(parstats2_2))
  col_true1_1 <- which(params1_1 == param_true1)
  col_true1_2 <- which(params1_2 == param_true2)
  col_true2_1 <- which(params2_1 == param_true1)
  col_true2_2 <- which(params2_2 == param_true2)

  len <- function(...) length(...)
  pdat <- data.frame(y=c(cov1_1, cov2_1, cov1_2, cov2_2), x=c(params1_1, params2_1, params1_2, params2_2),
                     legend=c(rep(c("cens1", "cens2"), c(len(cov1_1), length(cov2_1))),
                              rep(c("cens1", "cens2"), c(len(cov1_2), length(cov2_2)))),
                     parm=rep(c("par1", "par2"),
                              c(len(cov1_1)+length(cov2_1), len(cov1_2)+length(cov2_2))  ))
  pdat_lines <- data.frame(x=c(param_true1, param_true2), y=rep(95,2), parm=c("par1", "par2"))

  p <- ggplot() + labs(list(x="parameter value", y="coverage (%)"))

  p <- p + geom_vline(data=pdat_lines, aes(xintercept=x), col="red", linetype="longdash", size=2)
  p <- p + geom_hline(data=pdat_lines, aes(yintercept=y), linetype="dashed", size=1)

  p <- p + geom_path(data=pdat, aes(x=x, y=y, col=legend), size=1) +
    geom_point(data=pdat, aes(x=x, y=y, col=legend), size=3)

  p <- p + scale_y_continuous(breaks=100*c(0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1))
  p <- p + coord_cartesian(ylim=100*c(0.4,1.01))

  number_ticks <- function(n) {function(limits) pretty(limits, n)}
  if(!missing(limx))
    p <- p + scale_x_continuous(limits=limx, breaks=number_ticks(x_nbreaks))
  else
    p <- p + scale_x_continuous(breaks=number_ticks(x_nbreaks))

  ## faceting ##
  facet_names <- list("par1"=labx1, "par2"=labx2)
  facet_label_fun <- function(variable, value)
    return(facet_names[value])
  p <- p + facet_grid(.~parm, labeller=facet_label_fun, scales="free_x")

  p2 <- p + scale_colour_manual(values=c("green", "mediumseagreen"),
                                labels=c("0% censoring", "20% censoring"))
  p2 <- p2 + theme(legend.title=element_blank())

  if(show_plot){
    if(win_plot)
      windows(width, height)
    print(p2)

    if(!missing(fname))
      ggsave(file=fname)
  }

  return(invisible(p2))
}


cov_ggplot_prior <- function(sim1_1, sim1_2, sim2_1, sim2_2, sim3_1, sim3_2,
                             param_true, model, limx, x_nbreaks=6,
                             show_plot=TRUE, win_plot=TRUE, width=14, height=6, fname)
{
  if(model == "crgA"){
```

```
    labx   <- expression(paste(alpha, " parameter value"))
    p_ind <- 5
  }
  else if(model == "crgB"){
    labx   <- expression(paste(beta, " parameter value"))
    p_ind <- 6
  }
  else if(model == "gcrgA"){
    labx   <- expression(paste(alpha, " parameter value"))
    p_ind <- 7
  }
  else if(model == "gcrgB"){
    labx   <- expression(paste(beta, " parameter value"))
    p_ind <- 8
  }
  else if(model == "gcrgC"){
    labx   <- expression(paste(italic(c), " parameter value"))
    p_ind <- 9
  }
  else
    stop("Details incorrectly specified.")

  parstats1_1 <- sim1_1[[p_ind]] ; parstats1_2 <- sim1_2[[p_ind]]
  parstats2_1 <- sim2_1[[p_ind]] ; parstats2_2 <- sim2_2[[p_ind]]
  parstats3_1 <- sim3_1[[p_ind]] ; parstats3_2 <- sim3_2[[p_ind]]

  cov1_1  <- parstats1_1[3,]*100
  cov2_1  <- parstats2_1[3,]*100
  cov3_1  <- parstats3_1[3,]*100
  params1_1 <- as.numeric(colnames(parstats1_1))
  params2_1 <- as.numeric(colnames(parstats2_1))
  params3_1 <- as.numeric(colnames(parstats3_1))
  col_true1_1 <- which(params1_1 == param_true)
  col_true2_1 <- which(params2_1 == param_true)
  col_true3_1 <- which(params3_1 == param_true)

  cov1_2  <- parstats1_2[3,]*100
  cov2_2  <- parstats2_2[3,]*100
  cov3_2  <- parstats3_2[3,]*100
  params1_2 <- as.numeric(colnames(parstats1_2))
  params2_2 <- as.numeric(colnames(parstats2_2))
  params3_2 <- as.numeric(colnames(parstats3_2))
  col_true1_2 <- which(params1_2 == param_true)
  col_true2_2 <- which(params2_2 == param_true)
  col_true3_2 <- which(params3_2 == param_true)

  len <- function(...) length(...)
  pdat <- data.frame(y=c(cov1_1, cov2_1, cov3_1, cov1_2, cov2_2, cov3_2),
                     x=c(params1_1, params2_1, params3_1, params1_2, params2_2, params3_2),
                     legend=c(rep(c("prior1", "prior2", "prior3"), c(len(cov1_1), len(cov2_1), len(cov3_1))),
                     rep(c("prior1", "prior2", "prior3"), c(len(cov1_2), len(cov2_2), len(cov3_2)))),
                     cens_level=rep(c("cens1", "cens2"),
                     c(len(cov1_1)+len(cov2_1)+len(cov3_1), len(cov1_2)+len(cov2_2)+len(cov3_2))))
  pdat_lines <- data.frame(x=param_true, y=95)

  p <- ggplot() + labs(list(x=labx, y="coverage (%)"))

  p <- p + geom_vline(data=pdat_lines, aes(xintercept=x), col="red", linetype="longdash", size=2)
  p <- p + geom_hline(data=pdat_lines, aes(yintercept=y), linetype="dashed", size=1)

  p <- p + geom_path(data=pdat, aes(x=x, y=y, col=legend), size=1) +
    geom_point(data=pdat, aes(x=x, y=y, col=legend), size=3)

  p <- p + scale_y_continuous(breaks=100*c(0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1))
  p <- p + coord_cartesian(ylim=100*c(0.4,1.01))
```

```
984
985   number_ticks <- function(n) {function(limits) pretty(limits, n)}
986   if(!missing(limx))
987     p <- p + scale_x_continuous(limits=limx, breaks=number_ticks(x_nbreaks))
988   else
989     p <- p + scale_x_continuous(breaks=number_ticks(x_nbreaks))
990
991   ## faceting ##
992   facet_names <- list("cens1"="0% censored", "cens2"="20% censored")
993   facet_label_fun <- function(variable, value)
994     return(facet_names[value])
995   p <- p + facet_grid(.~cens_level, labeller=facet_label_fun)
996
997   p2 <- p + scale_colour_manual(values=c("green", "mediumseagreen", "forestgreen"),
998                                 labels=c("Jeffreys", "Reference", "PM"))
999   p2 <- p2 + theme(legend.title=element_blank())
1000
1001  if(show_plot){
1002    if(win_plot)
1003      windows(width, height)
1004    print(p2)
1005
1006    if(!missing(fname))
1007      ggsave(file=fname)
1008  }
1009
1010  return(invisible(p2))
1011 }
```

Bibliography

Abdel-Ghaly, A. and Attia, A. 1993. On a new density function. *Microelectronics and reliability*, 33(5):671–679.

Abushal, T. 2011. Estimation of the unknown parameters for the compound Rayleigh distribution based on progressive first-failure-censored sampling. *Open Journal of Statistics*, 1:161–171.

Ahmed, A., Ahmad, S., and Reshi, J. 2013. Bayesian analysis of Rayleigh distribution. *International Journal of Scientific and Research Publications*, 3(10).

Al-Kanani, I. and Jasim, S. 2014. Non-Bayesian and Bayesian estimation for generalized Rayleigh distribution. *International Journal of Modern Mathematical Sciences*, 10(2): 103–115.

Ando, T. 2011. Predictive Bayesian model selection. *American Journal of Mathematical and Management Sciences*, 31(1:2):13–38.

Andrews, M. and Baguley, T. 2013. Prior approval: the growth of Bayesian methods in psychology. *British Journal of Mathematical and Statistical Psychology*, 66(1):1–7.

Barot, D. and Patel, M. 2014. Risk efficiencies of empirical Bayes and generalized maximum likelihood estimates for Rayleigh model under censored data. *Journal of Probability and Statistics*, 2014.

Bayes, T. and Price, R. 1763. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53:370–418.

Beaumont, M. and Rannala, B. 2004. The Bayesian revolution in genetics. *Nature Reviews Genetics*, 5:251–261.

Bekker, A., Roux, J., and Mostert, P. 2000. A generalization of the compound Rayleigh distribution: using a Bayesian method on cancer survival times. *Communications in Statistics – Theory and Methods*, 29(7):1419–1433.

Berger, J. and Bernardo, J. 1989. Estimating a product of means: Bayesian analysis with reference priors. *Journal of the American Statistical Association*, 84(405):200–207.

Berger, J. and Bernardo, J. On the development of reference priors. In Bernardo, J., Berger, J., Dawid, A., and Smith, A., editors, *Bayesian Statistics 4*, pages 35–60. Oxford University Press, Oxford, 1992, pages 35–60.

Berliner, L. and Hill, B. 1988. Bayesian nonparametric survival analysis. *Journal of the American Statistical Association*, 83(403):772–779.

Bernardo, J. 1979. Reference posterior distributions for Bayesian inference. *Journal of the Royal Statistical Society, Series B*, 41(2):113–147.

Best, D., Rayner, J., and Thas, O. 2010. Easily applied tests of fit for the Rayleigh distribution. *Sankhya: The Indian Journal of Statistics, Series B*, 72(2):254–263.

Bhattacharya, S. and Tyagi, R. 1990. Bayesian survival analysis based on the Rayleigh model. *Trabajos De Estadistica*, 5(1):81–92.

Casas-Prat, M. and Holthuijsen, L. 2010. Short-term statistics of waves observed in deep water. *Journal of Geophysical Research: Oceans*, 115(C9).

Celeux, G., Forbes, F., Robert, C., and Titterington, D. 2006. Deviance information criteria for missing data models. *Bayesian Analysis*, 1:651–706.

Chen, M. and Shao, Q. 1999. Monte Carlo estimation of Bayesian credible and HPD intervals. *Journal of Computational and Graphical Statistics*, 8(1):69–92.

Colosimo, E., Ferreira, F., Oliveira, M., and Sousa, C. 2002. Empirical comparisons between Kaplan-Meier and Nelson-Aalen survival function estimators. *Journal of Statistical Computation and Simulation*, 72(4):299–308.

Cordeiro, G., Cristino, C., Hashimoto, E., and Ortega, E. 2013. The beta generalized Rayleigh distribution with applications to lifetime data. *Statistical Papers*, 54:133–161.

Cox, D. 1972. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B*, 34(2):187–220.

Das, S., Spall, J., and Ghanem, R. 2010. Efficient Monte Carlo computation of Fisher information matrix using prior information. *Computational Statistics and Data Analysis*, 54:272–289.

Datta, G. and Ghosh, J. 1995. On priors providing frequentist validity for Bayesian inference. *Biometrika*, 82(1):37–45.

Dellaportas, P. and Wright, D. 1991. Numerical prediction for the two-parameter Weibull distribution. *The Statistician*, 40:365–372.

Eberly, L. and Casella, G. 2003. Estimating Bayesian credible intervals. *Journal of Statistical Planning and Inference*, 112:115–132.

Fisher, R. 1925. Theory of statistical estimation. *Proceedings of the Cambrigde Philosophical Society*, 222:700–725.

Gelman, A. 2008. Objections to Bayesian statistics. *Bayesian Analysis*, 3(3):445–449.

Gelman, A. and Rubin, D. 1992. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472.

Gelman, A., Carlin, J., Stern, H., and Rubin, D. 2004. *Bayesian Data Analysis (2nd Edition)*. Chapman & Hall, New York.

Ghitany, M. 2001. A compound Rayleigh survival model and its application to randomly censored data. *Statistical Papers*, 42:437–450.

Gilks, W., Richardson, S., and Spiegelhalter, D. 1996. *Markov chain Monte Carlo in practice*. Chapman & Hall, Boca Raton, FL.

Gomes, A., Da-Silva, C., Cordeiro, G., and Ortega, E. 2014. A new lifetime model: the Kumaraswamy generalized Rayleigh distribution. *Journal of Statistical Computation and Simulation*, 84(2):290–309.

Greenwich, M. 1992. A unimodal hazard rate function and its failure distribution. *Statistical Papers*, 33(1):187–202.

Grubbstrom, R. and Tang, O. 2006. The moments and central moments of a compound distribution. *European Journal of Operational Research*, 170:106–119.

Gurland, J. 1957. Some interrelations among compound and generalized distributions. *Biometrika*, 44(1):265–268.

Guure, C., Ibrahim, N., and Ahmed, A. 2012. Bayesian estimation of two-parameter Weibull distribution using extension of Jeffreys' prior information with three loss functions. *Mathematical Problems in Engineering*, 2012(ID 589640).

Hastings, W. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Hill, B. 1987. The validity of the likelihood principle. *The American Statistician*, 41(2): 95–100.

Jaynes, E. T. *Baysian methods: General background*, page 1–25. Aug 1984.

Jeffreys, H. 1946. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London: A*, 186(1007).

Kaplan, E. and Meier, P. 1958. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53(282):457–481.

Klein, J. and Moeschberger, M. 2003. *Survival Analysis Techniques for censored and truncated data (2nd Edition)*. Springer-Verlag, New York.

Kronrod, A. 1965. *Nodes and weights of quadrature formulas*. Consultants Bureau, New York.

Kundu, D. and Raqab, M. 2005. Generalized Rayleigh distribution: different methods of estimations. *Computational Statistics and Data Analysis*, 49:187–200.

Kundu, D. and Raqab, M. 2015. Estimation of $R = P[Y < X]$ for three-parameter generalized Rayleigh distribution. *Journal of Statistical Computation and Simulation*, 85(4):725–739.

Lalitha, S. and Mishra, A. 1996. Modified maximum likelihood estimation for Rayleigh distribution. *Communications in Statistics – Theory and Methods*, 25(2):389–401.

Lord Rayleigh. 1880. On the resultant of a large number of vibrations of the same pitch and of arbitrary phase. *Philosophical Magazine Series 5*, 10(60):73–78.

Martz, H. and Waller, R. 1982. *Bayesian reliability analysis*. John Wiley & Sons, New York.

Matsumoto, M. and Nishimura, T. 1998. Mersenne Twister: a 632-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1):3–30.

McDonald, J. and Butler, R. 1987. Some generalized mixture distributions with an application to unemployment duration. *The Review of Economics and Statistics*, 69 (2):232–240.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. 1953. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092.

Mostert, P., Bekker, A., and Roux, J. 1998. Bayesian analysis of survival data using the Rayleigh model and LINEX loss. *South African Statistical Journal*, 32:19–42.

Mostert, P., Roux, J., and Bekker, A. 1999. Bayes estimators of the lifetime parameters using the compound Rayleigh model. *South African Statistical Journal*, 33:117–138.

Mudholkar, G., Srivastava, D., and Kollia, G. 1996. A generalization of the Weibull distribution with application to the analysis of survival data. *Journal of the American Statistical Association*, 91(436):1575–1583.

Piessens, R., De Doncker-Kapenga, E., Uberhuber, C., and Kahaner, D. 1983. *Quadpack: a Subroutine Package for Automatic Integration*. Springer-Verlag, Berlin.

Poirier, D. 2006. The growth of Bayesian methods in statistics and economics since 1970. *Bayesian Analysis*, 1(4):969–980.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012. URL `http://www.R-project.org/`.

Rivoirard, V. and Rousseau, J. 2012. Bernstein-von Mises theorem for linear functionals of the density. *The Annals of Statistics*, 40(3):1489–1523.

Rizzo, M. 2008. *Statistical computing with R*. Chapman & Hall, Boca Raton, FL.

Robert, C. 2007. *The Bayesian Choice (2nd Edition)*. Springer Science, New York.

Saleem, M. and Aslam, M. 2008. Bayesian analysis of the two component mixture of the Rayleigh distribution with uniform and Jeffreys priors. *Journal of Applied Statistical Science*, 16(4):105–113.

Saleem, M. and Aslam, M. 2009. On bayesian analysis of the Rayleigh survival time assuming the random censor time. *Pakistan Journal of Statistics*, 25(2):71–82.

Shannon, C. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423.

Siddiqui, M. 1962. Some problems connected with Rayleigh distributions. *J. Res. Nat. Bur. Stand., Radio Propagation*, 66D(2):167–174.

Sijbers, J., Den Dekker, A., Raman, E., and Van Dyck, D. 1999. Parameter estimation from magnitude MR images. *International Journal of Imaging Systems and Technology*, 10(2):109–114.

Soliman, A. 2006. Estimators for the mixture of Rayleigh model based on progressively censored data. *Communications in Statistics – Theory and Methods*, 35(5):803–820.

Spiegelhalter, D., Best, N., Carlin, B., and Van der Linde, A. 2002. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B*, 64(4): 583–639.

Spiegelhalter, D., Best, N., Carlin, B., and Van der Linde, A. 2014. The deviance information criterion: 12 years on. *Journal of the Royal Statistical Society: Series B*, 76(3):485–493.

Stablein, D., Carter, W., and Novak, J. 1981. Analysis of surival data with nonproportional hazard functions. *Controlled Clinical Trials*, 2:149–159.

Stigler, S. 1986. Laplace's 1774 memoir on inverse probability. *Statistical Science*, 1(3): 359–378.

Surles, J. and Padgett, W. 2001. Inference for reliability and stress-strength for a scaled Burr Type X distribution. *Lifetime Data Analysis*, 7:187–200.

Wackerley, D., Mendenhall, W., and Scheaffer, R. 2008. *Mathematical Statistics with applications, (7th edition)*. Thomson, Belmont, Canada.

Wynn, P. 1956. On a device for computing the $e_m(S_n)$ transformation. *Mathematical Tables and Other Aids to Computation*, 10(54):91–96.

Zellner, A. 1986. Bayesian estimation and prediction using asymmetric loss functions. *Journal of the American Statistical Association*, 81(394):446–451.